Unsupervised Relation Extraction for E-Learning Applications

Naveed Afzal

A thesis submitted in partial fulfilment of the
requirements of the University of Wolverhampton
for the degree of Doctor of Philosophy
2012

This work or any part thereof has not previously been presented in any form to the University or to any other institutional body whether for assessment, publication, or for other purposes (unless otherwise indicated). Save for any express acknowledgements, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

Signature………………….
Date………………..

Wisdom is not the product of schooling but the lifelong attempt to acquire it. (Albert Einstein)

Abstract

In this modern era many educational institutes and business organisations are adopting the e-Learning approach as it provides an effective method for educating and testing their students and staff. The continuous development in the area of information technology and increasing use of the internet has resulted in a huge global market and rapid growth for e-Learning. Multiple Choice Tests (MCTs) are a popular form of assessment and are quite frequently used by many e-Learning applications as they are well adapted to assessing factual, conceptual and procedural information. In this thesis, we present an alternative to the lengthy and time-consuming activity of developing MCTs by proposing a Natural Language Processing (NLP) based approach that relies on semantic relations extracted using Information Extraction to automatically generate MCTs.

Information Extraction (IE) is an NLP field used to recognise the most important entities present in a text, and the relations between those concepts, regardless of their surface realisations. In IE, text is processed at a semantic level that allows the partial representation of the meaning of a sentence to be produced. IE has two major subtasks: Named Entity Recognition (NER) and Relation Extraction (RE). In this work, we present two unsupervised RE approaches (surface-based and dependency-based). The aim of both approaches is to identify the most important semantic relations in a document without assigning explicit labels to them in order to ensure broad coverage, unrestricted to predefined types of relations.

In the surface-based approach, we examined different surface pattern types, each implementing different assumptions about the linguistic expression of semantic relations between named entities while in the dependency-based approach we explored how dependency relations based on dependency trees can be helpful in extracting relations between named entities. Our findings indicate that the presented approaches are capable of achieving high precision rates.

Our experiments make use of traditional, manually compiled corpora along with similar corpora automatically collected from the Web. We found that an automatically collected web corpus is still unable to ensure the same level of topic relevance as attained in manually compiled traditional corpora. Comparison between the surface-based and the dependency-based approaches revealed that the dependency-based approach performs better. Our research enabled us to automatically generate questions regarding the important concepts present in a domain by relying on unsupervised relation extraction approaches as extracted semantic relations allow us to identify key information in a sentence. The extracted patterns (semantic relations) are then automatically transformed into questions. In the surface-based approach, questions are automatically generated from sentences matched by the extracted surface-based semantic pattern which relies on a certain set of rules. Conversely, in the dependency-based approach questions are automatically generated by traversing the dependency tree of extracted sentence matched by the dependency-based semantic patterns.

The MCQ systems produced from these surface-based and dependency-based semantic patterns were extrinsically evaluated by two domain experts in terms of questions and distractors readability, usefulness of semantic relations, relevance, acceptability of questions and distractors and overall MCQ usability. The evaluation results revealed that the MCQ system based on dependency-based semantic relations performed better than the surface-based one. A major outcome of this work is an integrated system for MCQ generation that has been evaluated by potential end users.

Acknowledgements

First of all, I would like to thank the Almighty who has enabled me to complete this thesis. This thesis would not have been possible without help from a lot of the people and I would like to take this opportunity to thank them.

Special thanks to my director of studies, Ruslan Mitkov, my supervisors Viktor Pekar and Atefeh Farzindar for their continuous guidance, support and encouragement. I would like to express my special gratitude and appreciation to Alison Carminke and Erin Stokes who took the trouble of reading the final draft of my thesis and helped me improve it with their valuable comments. I would also like to thank Syed Amir Iqbal and Ruth Seal for their help and feedbacks during evaluation.

This thesis would not be the way it is without the valuable comments and suggestions from members of the Research Group in Computational Linguistics at the University of Wolverhampton. In alphabetical order they are Miranda Chong, Iustin Dornescu, Richard Evans, Le An Ha, Iustina IIisei, Natali Konstantinova, Georgiana Marsic, Constantin Orasan, Yvonne Skalban, Lucia Specia and Irina Temnikova.

I would like to express my special gratitude and appreciation to my former research advisor Mark Stevenson who introduced me to the world of Natural Language Processing. I still think fondly of my time as a postgraduate student spent working with him.

Finally, but most importantly, I would also like to thank my family and friends for their unstinting support, motivation and encouragement. My parents have always believed in me and encouraged me to strive for excellence in all that I do. There are no sufficient words to thank them for their help and everything they did for me.

Table of Contents

List of Tables

List of Figures

Abbreviations

BNC – British National Corpus

CHI – Chi-Square

FBQ – Fill-in-the-Blank Question

GUI – Graphical User Interface

ICT – Information and Communication Technology

IE – Information Extraction

IG – Information Gain

IGR – Information Gain Ratio

IR – Information Retrieval

LL – Log-Likelihood

MCQ – Multiple Choice Question

MCT – Multiple Choice Test

MI – Mutual Information

MT – Machine Translation

MUC – Message Understanding Conference

NE – Named Entity

NER – Named Entity Recognition

NLG – Natural Language Generation

NMI – Normalised Mutual Information

NLP – Natural Language Processing

PoS – Parts-of-Speech

RE – Relation Extraction

SC – Semantic Class

SVO – Subject Verb Object

VLE – Virtual Learning Environment

Chapter 1: Introduction

1.1 E-Learning

In the modern era of information technology many organisations and institutions offer diverse forms of training to their employees or learners and most of these training options utilise e-Learning. In the last two decades, e-Learning has seen an exponential growth mainly due to the development of the internet, which has made online materials accessible to more people than ever, allowing many corporations, educational institutes, governments and other organisations to use it in their training process. E-learning has also been referred to by different terms such as online learning, web-based training and computer-based training.

E-learning is fundamentally a learning process that is facilitated and supported by Information and Communications Technology (ICT). Learning objectives play a pivotal role in the design of any learning material as they help to design lessons which are easier for the learner to comprehend and the instructor to evaluate. The quality of e-Learning depends upon its contents and its delivery. The concept of e-Learning is growing at a rapid rate, since more and more people are using computers frequently in every field of life. E-learning has made a huge impact in the field of education as it has been exploited effectively in higher education to enhance the traditional forms of teaching and administration and students are more comfortable with e-Learning methods and e-Learning technologies. E-learning can be CD-ROM-based, network-based or internet-based and it can contain text, audio, video and a Virtual Learning Environment (VLE). A VLE is a software platform on which learning materials are assembled and made available. Distance education (in which the learner and the instructor are separated by space and/or time) has also provided a base for e-Learning development. It is delivered through a variety of learning resources e.g. learning guides and supplementary digital media. Currently many educational institutes use *blended learning*, a term used to describe education that combines on-campus and distance learning approaches. It includes conventional on-campus courses supplemented by some e-Learning. In order for e-Learning to be effective it must use reliable and easy-to-use technology.

E-learning also has a major impact in the industrial field. The ability to acquire new skills and knowledge is important for any professional in this fast-moving world. According to a survey report in 2008[1] the vast majority of public sector (82%) and 42% of private sector organisations used e-learning for the training of their employees. The global market for e-Learning is growing at a rapid rate as many business organisations and educational institutes are seeking to deliver their learning in a smarter and more cost-effective way. E-learning products have a huge market world-wide: the UK e-learning market alone was estimated at between £500m - £700m in 2009[2]. The future of e-Learning depends on the development of IT technologies.

1.2 Automatic Assessment in E-Learning

Automatic assessment is one of the main strengths of e-Learning. Assessment is a process used to test the acquired knowledge of a person on a specific topic/subject. According to Linn and Miller (2005), "Assessment is a general term that includes the full range of procedures used to gain information about student learning (observations, ratings of performance or projects, paper-and-pencil tests) and the formation of value judgements concerning learning progress." Assessment has a vital role to play in the areas of education and training as it determines whether or not learning objectives are being met. Educational institutes such as schools and universities conduct regular assessments of their students. Effective assessment aids teachers in analysing learning problems and progress, improving and enhancing their own performance and achieving and maintaining academic standards. Many organisations, both in public and private sectors also conduct regular assessments of their employees as well as job applicants. In many areas, such as health-care and law,

[1] http://www.cipd.co.uk/NR/rdonlyres/3A3AD4D6-F818-4231-863B-4848CE383B46/0/learningdevelopmentsurvey.pdf

[2] http://www.e-learningcentre.co.uk/Reviews_and_resources/Market_Size_Reports_/The_UK_e_learning_market_2009

specialists have to undertake compulsory assessment procedures in order to attain national qualifications and the right to practice their profession. The development and delivery of assessment materials, the analysis of their results and provision of feedback to numerous test takers is an extremely laborious and time consuming task. According to (Stiggins, 2001) most teachers in schools or higher education institutes often lack the knowledge and skills to create effective assessment materials. Moreover they are also unable to correctly interpret the assessment results in order to use them for future adaptation.

Automatic assessment in e-Learning provides immediate feedback, enables the instructor to ensure the continuous intellectual, social and physical development of the learner and moreover can also be linked to other computer-based or online materials. ICT-based assessment support technologies have been used for some time in different educational scenarios (see McFarlane 2001, 2002; Weller, 2002 for a review of ICT-based assessment support). The use of ICT-based assessment has many advantages when compared to paper-pencil testing and it is more appropriate for large-scale assessments (e.g. Ball et al., 2003; Abell et al., 2004; Scheuermann and Pereira, 2008). Moreover ICT-based assessments considerably lessen the amount of time and money spent on manually producing assessment exercises (Pollock et al., 2000). ICT has been widely used to help authorise and deliver assessments to students by software such as TRIADS and QuestionMark, and frameworks such as OLAAF. ICT has also been used in assessment scoring and feedback provision (e.g. Leacock and Chodorow, 2003; Higgins et al., 2004; Pulman and Sukkarieh, 2005). TOEFL (Test of English as a Foreign Language), GRE (Graduate Record Examinations) and GMAT (Graduate Management Admission Test) are examples of widely used ICT-based assessments.

1.3 Multiple-Choice Questions (MCQs)

Multiple choice questions (MCQs), also known as Multiple-choice tests (MCTs), provide a popular solution for large-scale assessments as they make it much easier for test-takers to take tests and for examiners to interpret their results. MCTs are frequently used in various fields (e.g. education, market research, elections and policies) and can effectively measure a learner's knowledge and understanding levels. The emergence of e-Learning has created even higher demand for MCTs as it is one of the most effective ways for an e-learner to get feedback. Multiple-choice tests (MCTs) are a form of objective assessment in which a user selects one answer from a set of alternative choices for a given question.

In the literature (see, e.g., Isaacs, 1994) the structure of a multiple choice question is described as follows. A multiple choice question is known as an *item*. The part of text which states the question is called *the stem* while the set of possible answers (correct and incorrect) are called *options*. The correct answer is called the *key* while incorrect answers are called *distractors*. Figure 1 shows an example of a multiple choice question.

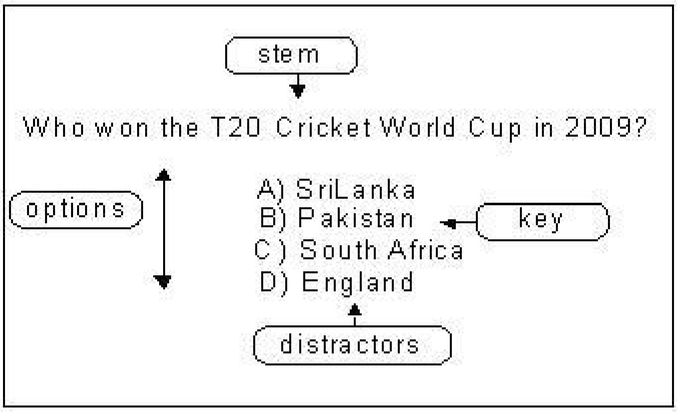


Figure 1: An example of a Multiple Choice Question

MCT items are close-ended questions and more suitable for assessing factual, conceptual and procedural information as they are straightforward to conduct and instantaneously provide an effective measure of test-takers' performance. MCTs have been employed by many instructors as a preferred assessment tool and it is estimated that 45% - 67% of student assessments utilise MCTs (Siegfried and Kennedy, 1995; Lister 2000, 2001; Becker and Watts, 2001 and Carter et al., 2003). MCTs lend themselves well to online delivery and computer grading. Most students are quite familiar with this mechanism of assessment. Usually an expert, trained in the relevant disciplines, is employed to create an MCT. The expert familiarises himself with all the reading materials examinees are supposed to know, designs questions and exercises relevant to the most vital concepts discussed in the materials and creates a list of possible answers.

MCTs face criticism due to the belief that they only test a superficial memorisation of facts and also that MCTs may be useful for formative assessment but they have no place in examinations where the student should be tested on more then just their ability to recall facts. Moreover, it requires substantial efforts to design the content of an MCT (McKeachie, 2002) as poorly written MCTs conceal learners' knowledge rather than revealing it (Becker and Johnston, 1999; Dufresne, Leonard and Gerace, 2002). The process of manually creating high quality MCTs is quite expensive in terms of time and resources. These costs become even higher when assessments are conducted at short intervals and the content of the test needs to be fresh for every session. Benton et al. (2004) presented a detailed analysis of MCT item generation, comparing MCT items generated with and without the aid of ICT. In ICT, WebCT™, a commercial course-management software package was used to deliver MCT items. Their experimental results revealed that the MCT items generated without the aid of ICT were poor and that ICT could really help instructors in creating better quality MCT items. They argued that MCT items generated with the aid of ICT would help instructors to achieve educational objectives by providing guidance and feedback for them to produce better quality MCT items in the future. Their study also affirmed the claims made by Stiggins (2001) that instructors do not know how to design effective assessments. Research has been carried out in order to determine the best ways to construct MCTs which can provide valid measures of target knowledge. Haladyna et al. (2002) conducted a literature review in the area of MCTs and presented a set of guidelines for the instructors to follow during manual construction MCTs.

1.4 Challenges in Automatic Generation of Multiple Choice Questions

In the previous section, we have discussed the definition of MCTs, their advantages, drawbacks and main guidelines to follow when writing MCTs (see Haladyna et al., 2002 for more details). In this section, we will look at the automatic generation of MCTs. As mentioned earlier, the main challenge in the construction of MCT items is the selection of important concepts in a document and the selection of plausible distractors which will enable confident test takers to be better distinguished from unconfident ones. Automated generation of MCT items would solve the problems faced during manual creation of MCT items. The objective of this research is to provide an alternative to the lengthy and laborious activity of developing MCT items by proposing a new automated approach for multiple choice questions (MCQs) generation.

All the recent approaches to automatically generate MCQs (see Section 2.1 for further details), in principle take input texts and generate questions by removing some words from a sentence, for example Mitkov et al. (2003, 2006) employed conversion patterns in order to convert declarative sentences into interrogatives. Their approach mainly relied on the use of a simple set of syntactic transformational rules in order to automatically generate questions. The methodology for distractors (wrong alternatives) varies from research to research. The main idea for distractor selection is to select semantically or compositionally similar words to the correct answer. Most of the studies use machine readable dictionaries for distractors selection. Mitkov et al. (2003, 2006) and Brown et al. (2005) employed WordNet[3], a lexical resource in which English nouns, verbs, adjectives and adverbs are grouped into synonym sets while Kunichika et al. (2002) and Sumita et al. (2005) used their in-house thesauri (see Section 2.1 for further details).

There are also a few commercial systems for effective delivery of learning materials such as MCQs. Questionmark[4] is a well-known and established leader in computerised education technologies in the world. Its products and services focus on technologies to facilitate remote, efficient and secure assessment of numerous test-takers. ETS[5] is a US-based, private non-profit organisation that provides assessment services around the world. ETS is the developer of the world-wide known TOEFL, SAT, and GRE tests. ETS also develops software tools for computerised MCTs

[3] an online lexical reference system by Princeton University. (http://wordnet.princeton.edu/)

[4] http://www.questionmark.com

[5] http://www.ets.org

assessment, which are similarly concerned with the management of test materials, their secure administration, analysis of their results and feedback to students.

In the field of Natural Language Processing (NLP), dealing with automatic generation of multiple choice questions is gaining a lot of attention since the last decade (see Section 2.1 for more details). NLP is a field in computer science and linguistics in which computers are used to process human languages in textual form in a way that is based on the meaning of the text in order to perform some useful task. The main motivation behind NLP is to build computer systems that can perform tasks which require understanding of textual language and to understand how humans communicate using language. Automatic generation of MCQs is an emerging topic in the application of NLP. In order to automatically generate MCQs it is important to identify important concepts and the relationships between those concepts in a text. NLP applications such as Term Extraction and Information Extraction help us to accomplish the aforementioned tasks. Automatic generation of questions can be considered as a specialised application of Natural Language Generation (NLG) which is a sub-area of NLP. The NLG task is to generate a natural language from a machine representing system such as a knowledge base or a logical form.

Recent advances in NLP technologies have enabled researchers to employ them in automatic generation of MCQs, but still the work done in this area does not have a long history. Most of the approaches (see Section 2.1 for further details) have extracted important concepts employing NLP technologies and transforming declarative sentences into questions. Some researchers (e.g. Brown et al., 2005; Hoshino and Nakagawa, 2005; Sumita et al., 2005) have employed automatically generated MCQs to measure test takers' proficiency in English. In recent times domain ontologies have also been employed to automatically generate MCQs.

1.5 Aims of the Thesis

The main aim of the thesis is to identify the ways in which Information Extraction (IE) methodologies can improve the quality of automatically generated MCT items and overcome the shortcomings faced by the previous approaches. Previous approaches (e.g. Mitkov et al., 2003, 2006 and Sumita et al., 2005) mostly rely on the syntactic structures of sentences to generate questions. The main problem with these approaches is the selection of appropriate sentences from which to automatically generate questions as sometimes a sentence is too simple or too complicated to be used. Therefore, in this research we will explore semantic relations between important concepts as processing of text at the semantic level allows us to produce the representation of the meaning of the sentence. The advantage of using a semantic relation is that it can be expressed using different syntactic structures. Semantic relations are the principal relations between two concepts expressed by words or phrases, e.g. Hypernymy (IS-A relation) and meronymy (Part-Whole relation). Semantic relations play a vital role in many NLP fields such as Information Extraction, Question Answering and Automatic Summarisation. Identification of semantic relations in a text is a complex task and it involves the discovery of certain linguistic patterns in the text that indicate the presence of a particular relation. A pattern consists of words and syntactic categories in text or the underlying syntactic structure (parse tree) of the text and a pattern represents the entities related by the semantic relation in the text. One of the drawbacks of the syntactic approach is that the wording of the question is similar to that of the original sentence (e.g. "Aspirin can relieve headaches."→ "Which of the following drugs can relieve headaches?"), hence it can be answered by somebody who tries to memorise complete sentences from the textbook. On the other hand, if the semantic relation between "aspirin" and "headache" can be established ("aspirin RELIEVE headache"), then patterns can be used to generate questions whose wordings do not depend on the original sentence wording. For example, if the relationship is "DRUG A RELIEVE SYMPTOM B" then the following templates for question can be used:

Which of the following drugs can relieve SYMPTOM B?
If you have SYMPTOM B, you should use which of the following drugs?

In this way, the generation engine would be more flexible and would be able to generate questions with different wordings.

1.6 Original Contributions

This thesis provides original contributions in the field of automatic generation of MCTs. This research presents a system for the automatic generation of MCT items based on Information Extraction methodologies as it is important to recognise the most important concepts present in a text and the relations between those concepts, regardless of their surface realisations. This research is mainly focused on generating MCTs from the biomedical domain but the presented approach is quite flexible and can easily be adapted to generate MCTs from other domains as well. Many NLP technologies which deliver promising results in the newswire or business domains do not yield good results in the biomedical domain (see Section 2.2 for further details). Moreover there is a lot of interest in techniques which can identify, extract, manage, integrate and discover new hidden knowledge from the biomedical domain.

In order to achieve this main aim, several goals need to be met. First of all, it is necessary to introduce the concept of IE, its major components and the important issues which need to be considered during the IE process. IE has two major components: Named Entity Recognition (NER) and Relation Extraction (RE). The thesis looks at various approaches (supervised, semi -supervised and unsupervised) for each component of IE. This thesis focuses on the RE component and investigates an unsupervised approach for RE as most of the recent IE approaches rely on some sort of domain-specific knowledge (e.g. seed examples, training data or hand-crafted rules, see Section 2.6 for more details) to extract relations from unannotated free text (e.g. Basili et al., 2000; Català et al., 2000; Harabagiu and Maiorano, 2000; Yangarber and Grishman, 2000; Yangarber 2000, 2003; Català, 2003, Greenwood et al., 2005; Stevenson and Greenwood, 2009) which is quite laborious and time-consuming. We employed an unsupervised RE approach as it allowed us to cover a potentially unrestricted range of semantic relations while most supervised and semi-supervised approaches can learn to extract only those relations that have been exemplified in annotated text, seed patterns or seed named entities. After the unsupervised RE process, important extracted semantic relations are then transformed into questions. The important issues which need to be considered during the question generation phase are the quality of generated questions and their syntactic correctness. After the question generation phase the generation of plausible distractors takes place. To assess the usefulness of the investigation, quality of the generated questions and distractors, an extrinsic evaluation is carried out. The system will be evaluated in terms of automatically generated questions for their readability, relevance, acceptability and usefulness of semantic relations and similarly automatically generated distractors will also be evaluated for their readability, relevance and acceptability. At the end, the overall acceptability of the whole automatically generated MCT items will also be assessed.

To summarise, the **original contributions** of this thesis are:

- Fully implemented automatically generated MCQ systems based on IE

- Adopted unsupervised Relation Extraction approaches (surface-based and dependency-based patterns) for the MCQs problem which extract important relations from the text.

- Various evaluation approaches to measure the association of extracted relations within the biomedical domain as compared to the general domain.

- Developed new methods for the generation of high quality questions which are grammatically and syntactically correct based on the extracted relations.

- Generation of plausible distractors for each question by utilising different semantic similarity measures.

- An extrinsic evaluation of automatically generated multiple choice test items.

1.7 System Overview

The overall architecture of the proposed system mainly consists of three modules: IE, question generation and distractor generation (see Figure 2). In order to automatically generate MCTs our research will focus on the following main steps: first we will recognise the important concepts in the text and the semantic relations between them using Information Extraction (IE) methodologies (Chapter 3). The extracted semantic relations will allow us to select the most appropriate sentences for automatic question generation. In later stages (Chapter 4) the extracted semantic relations will be transformed into questions by employing certain set of rules. The process of selecting plausible distractors will make use of a distributional similarity measure (Chapter 4).

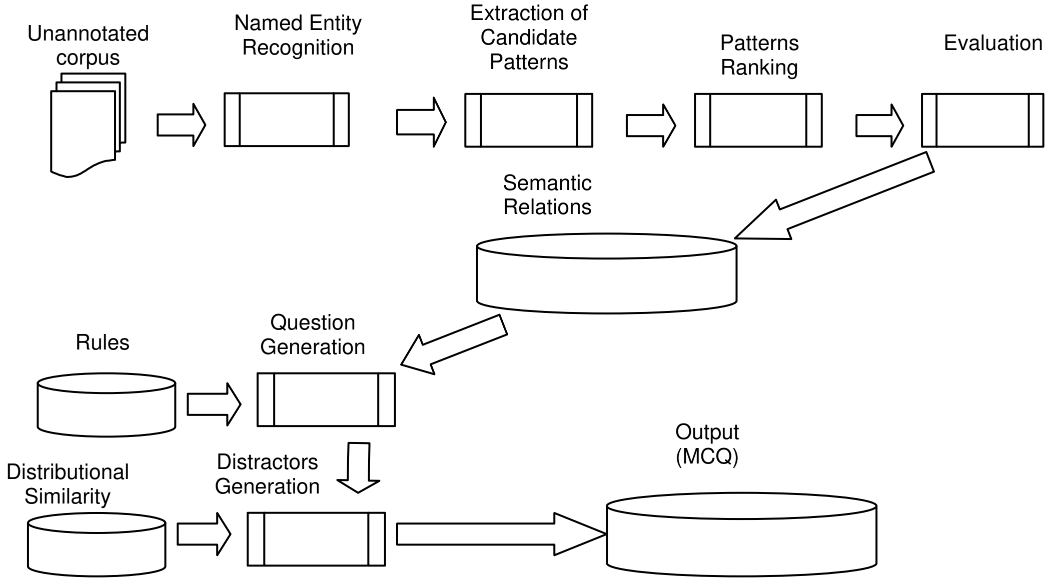


Figure 2: Overall system architecture

As mentioned earlier, Haladyna et al. (2002) proposed a set of guidelines for instructors to follow during the manual construction of MCTs in order to produce more effective and valid MCTs. These empirical guidelines address various issues during the manual construction of MCT items such as their readability, content, usability and effectiveness. Moreover, these guidelines emphasised that during the construction of MCTs instructors should focus on important concepts to test a higher level of learning, MCTs should not be too general, should be grammatically correct, should use simple vocabulary, must contain a single right answer and should make all distractors plausible. Our research will also follow these guidelines to automatically generate high-quality and effective MCT items. The use of semantic relations in our research will enable us to generate better quality MCT items by focusing on important concepts in the text while plausible distractors will be automatically generated using the distributional similarity measure.

1.8 Structure of the Thesis

The rest of this thesis is structured as follows: Chapter 2 provides the background for the automatic generation of MCT items and IE. Chapter 3 discusses the unsupervised approaches for relation extraction based on surface form and dependency trees, their evaluation in order to select stem sentences for the automatic generation of MCQs. Chapter 4 elaborates on the process of question generation and distractor generation while chapter 5 presents the extrinsic evaluation of the automatically generated MCT items. Chapter 6 contains the concluding remarks and future directions of work. In this section we elaborate the various tasks performed by each chapter in this thesis.

Chapter 2 provides the summary of the work done so far in the area of automatic generation of MCT items. This chapter then discusses the field of Information Extraction (IE), applications of IE, subtasks of IE, its two major components: Named Entity Recognition (NER) and Relation Extraction (RE), various supervised and unsupervised approaches for these components, evaluation of IE systems and various supervised, semi-supervised and unsupervised IE systems. In this chapter we look at the various dependency tree based pattern models and the comparison among these models. At the end of this chapter we also describe the use of Web as corpus.

Chapter 3 discusses unsupervised semantic relations extracted using IE techniques for stem sentences selection. It elaborates on two unsupervised approaches (surface-based and dependency-based) for RE from the biomedical domain. In the surface-based approach, we explore several different types of linguistic patterns while the dependency-based approach makes use of a slightly modified version of the linked chain model. Different pattern ranking methods (information theoretic and statistical) are used to rank the extracted patterns. We employed two different approaches to select the extracted patterns. The chapter ends by making a comparison between two unsupervised approaches.

Chapter 4 describes how extracted semantic relations in the form of linguistic patterns are used to select stem sentences and how these patterns are then transformed into syntactically correct automatically generated questions. Moreover, this chapter explains the different distributional similarity measures used to select plausible distractors for the automatically generated questions.

Chapter 5 presents an extrinsic evaluation of the whole MCT system in terms of question and distractor readability, relevance, usefulness of semantic relation and acceptability. At the end we also look at the overall usability of automatically generated MCT items.

Chapter 6 contains the concluding remarks and directions for future work.

Chapter 2: Background

In this chapter, we will discuss work done so far in the area of automatic generation of multiple choice test items. After that we will review previous work on NLP methods on which our own work draws in order to develop a new, semantics-aware method for automatic generation of MCQs. This chapter will present an overview of Information Extraction, its application in the real world and its two major components: Named Entity Recognition and Relation Extraction. This chapter will also provide a survey of the various supervised, semi-supervised and unsupervised approaches to building Information Extraction systems. We will also examine and compare various dependency tree based pattern models along with the use of the Web as a corpus.

2.1 Automatic Multiple Choice Question Generation

Even though NLP has made significant progress in recent years, NLP methods, and the area of automatic generation of MCT items in particular, have started being used in e-Learning applications only very recently.

One of the first significant studies in this area was published by Mitkov et al. (2003, 2006), who presented a computer-aided system for the automatic generation of multiple choice test items. Their system offered an alternative to the lengthy and demanding activity of manual construction of MCT items by proposing an NLP-based methodology for automatic generation of MCT items from instructive texts such as textbook chapters and encyclopaedia entries. Their system mainly consists of three parts: term extraction, stem generation and distractor selection. In the term extraction phase (Ha, 2007); the source text is parsed by a parser. The parser labelled each word in a source text with its part-of-speech and syntactic category. After the part-of-speech identification, nouns are sorted by their frequencies. The system employs certain rules and frequency thresholds for each noun and if any noun exceeds that threshold then that noun is regarded as a key term. The key terms are used to identify important concepts in a text from which questions are automatically generated. The key terms are domain-specific terms that will serve as the answers for the items. In the stem generation phase, stems are generated from the eligible clauses of sentences from the source text. A clause is considered eligible if it is finite and has SVO (Subject-Verb-Object) or SV (Subject-Verb) structure. The system makes use of several rules in order to generate a stem and to ensure grammaticality between the stem, the answer and the distractors. In order to produce plausible distractors, the system uses WordNet and retrieves hypernyms and coordinates of key terms from WordNet. The system was tested using a linguistic textbook in order to generate MCT items and found that 57% automatically generated MCT items were judged worthy of keeping as test items, of which 94% required some level of post-editing. The main advantage of this approach is that it has given a completely new alternative solution to the time-consuming and laborious activity of manual construction of MCT items, which is at the present moment the most extensively, used method for the students' knowledge evaluation. The main disadvantage of this system is its reliance on the syntactic structure of sentences to produce MCT items as it produces questions from sentences which have SVO or SV structure. Moreover, the identification of key terms in a sentence is also an issue as identification of irrelevant concepts (key terms) results in unusable stem generation.

Karamanis et al. (2006) conducted a pilot study to use Mitkov et al. (2006) system in a medical domain and their results revealed that some questions were simply too vague or too basic to be employed as MCQ in a medical domain. They concluded that further research is needed regarding question quality and usability criteria.

Skalban (2009) presented a detailed analysis of the Mitkov et al. (2006) system and highlighted the short-comings it faced. Her work distinguishes between critical and non-critical errors identified in the system output. Non-critical errors are errors with a low impact on the overall worthiness of the item; questions containing non-critical errors can typically be used after post-editing. Critical errors, however, have a detrimental impact on the worthiness of a question; post-editing is not possible. Her work also revealed that key term errors created the most unusable MCT items, accounting for nearly 50% of unworthy items. A key term error occurs, where a question has been generated based on a term which does not represent an important concept in the source text. On the surface, these questions can be syntactically flawless. However, they are still unworthy because questions generated from unimportant concepts are not useful for knowledge assessment.

Sumita et al. (2005) presented a system which automatically generated questions in order to measure test-takers' proficiency in English. The method described in this paper generates Fill-in-the-Blank Questions (FBQs) using a corpus, a thesaurus and the Web. The FBQs are created by replacing verbs with gaps in an input sentence. The possible distractors are retrieved from a thesaurus and then new sentences are created by replacing each gap in the input sentence with a distractor. They conducted their experiments on non-native speakers of the English Language and found that their method is quite effective in measuring proficiency of English in non-native speakers. The main drawback of this approach is that the selection of wrong input sentences results in FBQs which even native speakers are unable to answer. Moreover, the quality of generated FBQs is evaluated by a single native English speaker and it needs to be evaluated further.

Brown et al. (2005) used an approach that tests knowledge of students by automatically generating test items for vocabulary assessment. Their system produced six different types of questions for vocabulary assessment by making use of a WordNet. The six different types of questions include: definition, synonym, antonym, hypernym, hyponym and cloze questions. The *cloze question* requires the use of a target word in a specific context. In order to produce the *definition questions*, the system made use of the WordNet glosses to choose the first definition which did not include the target word. In *synonym questions*, it requires the matching of a target word to its synonym, which is extracted from WordNet. An *antonym question* requires a word to match its antonym which is also obtained from WordNet while in *hypernym and hyponym questions* require the matching of a word to its hypernym and hyponym respectively. In order to produce cloze questions the system made use of the WordNet glosses. The experimental results suggested that automatically generated questions produced using this approach provides an efficient way to automatically assess word knowledge. The approach presented in this paper relied heavily on WordNet and is unable to produce any questions for words which are not present in WordNet.

Chen et al. (2006) presented an approach for the semi-automatic generation of grammar test items by employing NLP techniques. Their approach was based on manually designed patterns which were further used to find authentic sentences from the Web and were then transformed into grammatical test items. Distractors were also obtained from the Web with some modifications in manually designed patterns e.g. changing part-of-speech, adding, deleting, replacing or reordering of words. The experimental results of this approach revealed that 77% of the generated MCQs were regarded as worthy (i.e. can be used directly or needed only minor revision). The disadvantage of this approach is that it requires a considerable amount of effort and knowledge to manually design patterns which can later be employed by the system to generate grammatical test items.

A semi-automatic system to assist teachers to produce cloze tests based on online news articles was presented by Hoshino and Nakagawa (2007). In cloze tests, questions are generated by removing one or more words from a passage and the test takers have to fill in the missing words. According to this paper, one of the reasons for selecting newspaper articles is that they are usually grammatically correct and suitable for English education. The system focuses on multiple-choice fill-in-the-blank tests and generates two types of distractors: vocabulary distractors and grammar distractors. For vocabulary distractors the system employs a frequency-based method while for grammar distractors the system makes use of ten grammar targets based on Tateno's (2005) research. The system mainly consists of two components: pre-processed component and graphical user interface (GUI). The input documents are first pre-processed and then go through various sub-processes which include: text extraction, sentence splitting, tagging and lemmatisation, synonym lookup, frequency annotation, inflection generation, grammar target mark-up, grammar distractor generation and selection of vocabulary distractors. The GUI allows the user to interact with the system. User evaluation reveals that 80% of the generated items were deemed to be suitable.

A system for automatic generation of MCT items which makes use of domain ontologies was presented by Papasalouros et al. (2008). Ontologies contain the domain knowledge of important concepts and relationships among these concepts. Ontologies contain knowledge which can be inferred, i.e. facts which are not explicitly defined. In order to generate MCTs, this paper utilised three different strategies: class-based strategies (based on hierarchies), property-based strategies (based on roles between individuals) and terminology-based strategies. The MCTs generated by this approach were evaluated in terms of quality, syntactic correctness and number of questions produced for different domain specific ontologies. The experimental results revealed that not all questions produced are syntactically correct and in order to overcome this problem more sophisticated Natural Language Generation (NLG) techniques are required. Moreover, property-based strategies produced a greater number of questions than class-based and terminology-based strategies but the questions produced by the property-based strategies are difficult to manipulate syntactically.

Most of the previous approaches to automatically generating MCTs have been used for vocabulary and grammatical assessments of English. Fundamentally most of the approaches generate questions by replacing some words from input text and mostly relies on syntactic transformations (e.g. Mitkov et al. 2003, 2006), generating questions by transforming declarative sentences into questions. The main drawback of these approaches is that generated MCTs are mostly based on recalling facts, grammatically correct but unusable in real life applications, so the main challenge is to automatically generate MCTs which will allow the examiner/instructor to evaluate test takers not only on superficial memorisation of facts but also on higher levels of cognition. This research solves this problem by extracting semantic rather than surface-level or syntactic relations between key concepts in a text via IE methodologies and then generating questions from such semantic relations. The methodology presented in this research will be unsupervised and can easily be adapted to other domains. In the next section we will discuss in detail the concept of IE and various approaches to IE.

2.2 Information Extraction (IE)

Information Extraction (IE) is an NLP field which is used to process unstructured natural language text and present it in a structured form such as a database. IE is the identification of specific items of information from text. The goal of IE is to extract salient facts about pre-specified types of semantic classes of objects (entities) and relationships among these entities. Entities are generally noun phrases in unstructured text e.g. names of persons, posts, locations and organisations, while relationships between two or more entities are described in a pre-defined way e.g. “interact with” is a relationship between two biological objects (proteins). This extracted information is then automatically stored into databases in order to be used for further processing. A pattern matching approach is usually employed by many IE systems where each pattern consists of a regular expression and an associated mapping from syntactic to logical form. During the pattern extraction process it is important to extract patterns that are general enough to extract correct information from the text but at the same time make sure that they do not extract incorrect information.

For example

“James Anderson was appointed vice president of the Proctor & Gamble Company of London”.

In the above mentioned example the entities we are interested in extracting are underlined and these are:

Person = James Anderson
Company = Proctor & Gamble
Post = Vice President.

Generally, a template is used to define the items of interest in a specific text. A template consists of a collection of slots (e.g. in the aforementioned example these slots are Person, Company and Post), each of which may be filled with one or more values.

Portability is one of the major issues in IE as adapting an existing IE system to a new domain requires manual tuning of domain-independent linguistic knowledge such as terminological dictionaries, domain-specific lexico-semantics, and extraction patterns and so on. Building these domain-independent linguistic knowledge resources by hand is very laborious and time-consuming, so automatic methods using NLP are required to learn them. Apart from portability, the large-scale IE systems also face many other challenges in terms of achieving high accuracy, performance, maintainability and usability (see Feldman, 2006 for further details).

2.2.1 Applications of IE

IE is widely used in many applications. It is utilised to automatically track specific event types from news sources and tracking disease outbreaks (Grishman et al., 2002). Many customer-oriented organisations collect many forms of unstructured data from customer interactions. In order to make effective use of this data, IE is applied to integrate this data with organisational databases. IE also has a great deal of information to offer to end-user industries of all kinds, mainly banks, financial companies, publishers and governments. For example, finance companies would really be interested to know: which company's acquisition took place in a specified time span; they would actually like to have widely spread text information compressed into a simple database.

IE is used in Personal Information Management (PIM) systems which seek to organise personal data like personal information, emails, personal activities, projects and people in a structured inter-linked format (Cai et al., 2005; Chakrabarti et al., 2005; Cutrell and Dumais, 2006).

There is a lot of research being done in the area of bio-informatics recently and a major problem in this area is extraction of biological objects and relationships between them from repositories e.g. extraction of protein names and their interaction from PubMed[6] (Bunescu et al., 2005; Plake et al., 2006). Moreover, IE has been successfully playing its part in the processing of clinical documents including patient discharge summaries, radiology reports and in assisting clinical decisions (Harkema et al., 2005; Savova et al., 2008; Boytcheva et al., 2009).

[6] http://www.ncbi.nlm.nih.gov/pubmed/

Many web-oriented applications make frequent use of IE. Many citation web databases such as Citeseer[7] and Google Scholar[8] employ IE in order to extract individual publication records, title, authors, references from papers and segmenting citation strings into individual authors, title, venue and year fields (Ponomareva et al., 2009). IE is used for automatic annotation of web pages for the semantic web (Stevenson and Ciravegna, 2003). IE is also applied to build opinion databases from blogs, newsgroup posts and product reviews which in turn help organisations to find out useful features of a product and widespread polarity of opinion regarding a specific product (Liu et al., 2005; Popescu and Etzioni, 2005).

Moreover, IE also interacts with many other areas of NLP including text classification, information retrieval, text mining and question answering (Ravichandran and Hovy, 2002). For example IE in a multi-lingual NLP environment may help a machine translation system to translate important facts accurately into the source language as it can provide the knowledge base for information retrieval, question answering and text summarisation (Heng, 2008). IE can also help to improve the performance of a text mining system by discovering useful knowledge from unstructured text (Mooney and Bunescu, 2005).

2.2.2 Subtasks of IE

The process of IE generally consists of the following subtasks (see Jurafsky and Martin, 2008 for more details):

Named Entity Recognition (NER): IE task which detects and classifies the proper names mentioned in a text

Co-reference resolution: links or clusters all the mentions that refer to the same named entity

Relation detection and classification/ Relation extraction: finds and classifies relations among the entities discovered in a given text

[7] http://citeseer.ist.psu.edu/

[8] http://scholar.google.co.uk/

Event detection and classification: finds events and fills in their participant slots with named entities detected

Temporal expression recognition: identifies temporal expressions in text

Temporal analysis: maps temporal expressions into specific dates or times of day

Template filling: fills in templates using snippets of text extracted from a given text or inferred from the text

Most of the aforementioned IE subtasks are domain dependent. In this research we will be focusing on the following two subtasks:

- Named Entity Recognition (NER)
- Relation Extraction (RE)

Named entity recognition (NER) is a key part of the IE system. NER involves identification of proper names in texts and classification into a set of predefined categories of interest. These Named Entities (NEs) will be different according to the nature of the text. For example: newspaper texts will contain the names of people, places and organisations while biological articles will contain the names of genes and proteins. Robust handling of proper names is an essential part of many NLP fields e.g. IR. A large amount of research has been done in NER in the recent past. There have been many main conference tracks and workshops on the topic of NER since 2000. Most of the early systems use handcrafted rule-based algorithms for NER while most of the modern systems employ various machine learning algorithms. The first major event dedicated to the NER task was in MUC-6 (Grishman and Sundheim, 1996). Two shared tasks for NER had been conducted with-in the conference on Computational Natural Language Learning (CoNLL): CoNLL 2002[9] (Tjong Kim Sang, 2000) and CoNLL 2003[10] (Tjong Kim Sang and Meulder, 2003). Several NER systems (Nadeau and Sekine, 2007) were developed to address diverse textual genres and domains, for example; Maynard et.al (2001) designed a system for emails, specific texts and religious texts. Porting an existing NER system to a new domain or textual genre still remains a major challenge.

[9] http://www.cnts.ua.ac.be/conll2002/ner/

[10] http://www.cnts.ua.ac.be/conll2003/ner/

Following NER the next step is the RE phase. The goal is to identify all the instances of specific relationships or events in text. For example, it is not just sufficient to find the occurrence of two biological objects (e.g. protein, gene) in a biomedical text but also to identify if there is a relationship between those biological objects. Generally, a template is used to classify the items which are to be extracted from the text.

2.2.3 Evaluation of IE Systems

Information Extraction systems are normally evaluated by comparing the performance of a system against the human judgement of the same text. The output that is identified by the humans is known as the *gold-standard*. IE system evaluations began with the Message Understanding Conferences (MUCs), which were sponsored by the U.S. government. These conferences were funded by the Defence Advanced Research Projects Agency (DARPA). One of the purposes of these conferences was to develop methods for the formal evaluation of IE systems (Grishman and Sundheim, 1996). Until now 7 Message Understanding Conferences (MUCs) have taken place and a different domain was selected for each conference. MUC-1 (1987) and MUC-2 (1989) were related to messages about naval operations. MUC-3 (1991) and MUC-4 (1992) were about news articles related to terrorist activities. MUC-5 (1993) was about news articles related to joint ventures and microelectronics. MUC-6 (1995) was about news articles related to management changes while MUC-7 (1997) was about news articles related to space vehicles and missile launches. Automatic Content Extraction (ACE)[11] evaluation has carried forward the work that was started by MUCs conferences by organising various evaluation tasks. ACE tasks include named entity detection and recognition, relation detection and recognition, event relation detection and recognition, co-reference resolution and named entity translation. Text Analysis Conference (TAC)[12] has held a series of evaluations and workshops to provide an infrastructure for large-scale evaluation of different NLP fields (e.g. question answering, recognising textual entailment, summarisation and knowledge base populations).

[11] http://www.itl.nist.gov/iad/mig//tests/ace/

[12] http://www.nist.gov/tac/about/index.html

The main aim of evaluation is to find out whether the system can identify the output in the gold-standards and not the extra ones. IE lends Information Retrieval (IR) concepts of Precision and Recall for evaluation. A system's Precision score is used to measure the number of relations identified that are correct while Recall score measures the number of correct relations that were identified.

Precision (P) = Correct Answers / Answers Produced
Recall (R) = Correct Answers / Total Possible Correct

Both notions can be made clear by examining the contingency table (Table 1):

	Correct (System)	*Incorrect (System)*
Correct (Gold Standard)	True Positives (TP)	False Positives (FP)
Incorrect (Gold Standard)	False Negatives (FN)	True Negatives (TN)

Table 1: Contingency table

True Positives (TP) are the correct answers produced by the system while False Positives (FP) are answers produced by the system which are not present in the gold-standard. False Negatives (FN), correct answers present in the gold-standard but not identified by the system while True Negatives (TN) are incorrect answers identified by both the gold-standard and the system.

$$P = \frac{TP}{(TP + FP)}$$

$$R = \frac{TP}{(TP + FN)}$$

Precision ranges between 0 (none of the identified events were correct) and 1 (all of them were correct) while Recall also ranges between 0 (no correct events identified) and 1 (all of the correct events were identified).

Precision and Recall is often combined into a single metric: F-measure, which is the harmonic mean of precision and recall.

$$F = \frac{2PR}{(P + R)}$$

In the aforementioned equation of F-measure both Precision and Recall are given equal weights. Precision and Recall are inversely proportional to each other which means that it is possible to boost one at the cost of reducing the other depending on the needs of the indented application. For example, an IR system (e.g. search engine) can often increase its Recall by retrieving more documents at the cost of increasing number of irrelevant documents retrieved (decreasing Precision).

Another alternative to judge an IE or IR system is its Accuracy, that is, the fraction of its classifications (correct and incorrect in IE while relevant and irrelevant in IR) that are correct. In terms of the contingency table (Table 1) Accuracy of a system is identified as:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Accuracy is not considered an appropriate measure of evaluation in either IR or IE due to data skewedness (see Manning et al., 2008 for further details). The measures of Precision and Recall are preferred as both concentrate on the return of True Positives (TP), asking what percentage of correct answers has been found by the system and how many False Positives (FP) have also been returned by the system.

In supervised approaches (see Section 2.2.5), in order to evaluate the performance of a classifier the data set is usually divided into three independent parts: the training data, the validation data and the test data. Classifiers used the training data for learning, the validation data for parameter optimisation and the test data to calculate the error rate. Generally, most classifiers used one-third of the data for testing and the remaining two-thirds for training. In situations where training or testing data is not representative enough to cover all classes in the data then a statistical technique known as cross-validation is employed. In cross-validation data is divided into fixed number of folds of equal size and each fold in turn is used for testing and the remainder is used for training. 10-fold cross-validation has become the method mostly used in practical terms. In 10-fold cross-validation data is divided randomly into 10 parts and each part (fold) in turn is used for testing and the remainder for training and this procedure is repeated 10 times. The error rate is calculated each time and finally the 10 error estimates are averaged to obtain an overall error estimate. Lavelli et al. (2004) critically reviewed various evaluation methodologies used by various IE systems and emphasised the need for the development of more reliable and detailed evaluation methodology.

2.2.4 Strategies to Perform IE

There are a number of factors that influence the decision to utilise a particular strategy to build an IE system. These factors include: availability of training data, availability of linguistic resources, availability of knowledge engineers and the level of desired performance (see Kaiser and Milksch, 2005 for more details).

Generally, there are two strategies to build IE systems:

- Knowledge Engineering
- Statistical or Machine Learning

Most of the early IE systems (e.g. Lehnert et al., 1992; Riloff, 1993) were based on the knowledge engineering strategy but have suffered from a knowledge acquisition bottleneck. In the knowledge engineering strategy a human expert (a person who is familiar with the domain) defines hand-coded rules or regular expressions to perform the task of extracting desired information from the text. In order to achieve this goal, the human expert needs to have a decent linguistic understanding of the task in hand. This strategy is quite laborious and time-consuming as it depends highly on a domain-specific dictionary and therefore requires a great deal of manual engineering. The advantage of this strategy is that with sufficient skills and experience, high-precision systems can be developed. The disadvantages of this strategy are that it has a very meticulous development process and needs experts who have good knowledge and both linguistic and domain expertise. The systems built using this strategy generally have a low coverage/recall because it is very hard to ensure this using introspection alone, while manual analysis of a corpus is also very expensive and cannot guarantee adequate coverage either. This strategy is most suitable in scenarios where training data is scarce or expensive to acquire and the highest possible performance is critical.

The machine learning strategy mostly uses statistical methods and learns extraction patterns or rules from annotated corpora and interaction with users. The machine learning strategy is more centred on producing training data rather than hand-crafted rules as is the case in knowledge engineering strategy. Corpus statistics are then derived automatically from the training data and used to process novel data. The advantages of this strategy are domain portability, no need for a human expert and data-driven rules ensuring full coverage of examples. The disadvantage of this strategy is that it will not work if there is no training data (or only a small quantity). This strategy is most appropriate in situations where training data is available in large quantities and easy to obtain and where no skilled rule writers are available for the task. In order to achieve high accuracy, this strategy relies heavily on a large set of training examples. Statistical and machine learning approaches in the last few years have become quite popular among the IE research community (e.g. Soderland and Lehnert, 1994; Bikel et al., 1998; Kleinberg, 2002; McCallum and Jensen, 2003 and Wang et al., 2005).

2.2.5 Machine Learning Approaches in IE

In the last section, we introduced knowledge engineering and machine learning strategies in IE; in this section we will discuss various machine learning approaches used in IE. Since 2000, machine learning algorithms have been used quite frequently for building IE systems (Nadeau and Sekine, 2007). There are three main types of machine learning algorithms with respect to the degree of supervision they require:

- Supervised Algorithms
- Semi-supervised Algorithms
- Unsupervised Algorithms

Supervised approaches in IE exploit a procedure known as classification. Classification is the process of assigning objects from a universe to two or more classes. In a classification task, each input is considered in isolation from all other inputs and the set of labels is defined in advance. The classifier's performance is measured in terms of the error rate. If a classifier predicts the class of an object correctly then it is counted as success and error otherwise. In Supervised learning algorithms the system is given examples of text manually marked up (annotated) with what should be learned from it (e.g. NEs or relations). The focal point in supervised learning is to study the features of positive and negative examples over a large annotated corpus and devise rules that capture instances of a desired type. Supervised approaches have the advantage of having access to training data (containing positive and negative examples) which enables them to learn complex patterns and give good performance but the annotation of text with entities or events is a very time-consuming task. The annotation process is quite slow and it is difficult to set guidelines that cover every instance, but without proper guidelines data will be inconsistent. Classifiers use supervised learning in order to sort data into pre-defined groups. Many researchers have effectively used supervised learning for IE (e.g. Zelenko et al., 2003; Culotta and Sorensen, 2004; Bunescu and Mooney, 2006). One example of a supervised learning algorithm in IE is WHISK (Soderland, 1999) discussed in Section 2.5.3.

Semi-supervised learning algorithms require a small degree of supervision and utilise a technique called "bootstrapping" which uses a small set of seeds (examples) in order to start the learning process. The system then searches for sentences that contain these seed examples and tries to identify some contextual clues they have in common. The system then identifies other instances that appear in a similar context, adds them to the seed examples and starts the learning process again. This process continues until enough instances are gathered. In this approach very few examples of annotated text are specified and a large quantity of raw text (Ando and Zhang, 2005; Bunescu and

Mooney, 2007). The idea of using bootstrapping for IE pattern acquisition was first introduced by Riloff (1996). The examples of semi-supervised learning algorithms based on dependency trees used for pattern learning in IE are the work carried out by Yangarber et al. (2000) and Stevenson and Greenwood (2005) (see Section 2.6.3 for more details). Semi-supervised approaches result in a reduction of time and effort to manually produce hand crafting rules or patterns but it also has some drawbacks. The main disadvantage of semi-supervised approaches is that though seed examples could be very reliable for a given task, the accuracy of the learned patterns decreases dramatically if any wrong patterns are accepted during the iteration process. Moreover, semi-supervised approaches are dependant on the set of seed examples provided by the expert as a bad set of seed examples could lead to a poor set of extraction patterns.

Unsupervised learning algorithms do not rely on any hand-labelled training data or seed examples. Most of the unsupervised learning algorithms use a technique called “clustering”. The process of clustering organises similar set of observations (patterns in our case) into small subsets known as clusters. Unsupervised learning algorithms are mostly used in scenarios where annotated data or seed examples are not available. Both classification and clustering place objects into groups or classes but the major difference between classification (supervised learning) and clustering (unsupervised learning) is that in the classification process classes are pre-defined while in the clustering process nothing is defined in advance. Examples of unsupervised learning algorithms applied in IE include Sekine (2006); Shinyama and Sekine (2006) and Eichler et al. (2008) (discussed in detail in Section 2.7).

2.3 Approaches to building Named Entity Recognition Systems

The first systems for NER were rule-based, based on pattern matching rules and pre-compiled lists of information i.e. gazetteers, the research community has since moved towards machine learning methods for NER. For example, in the MUC-7 competition[13] five NER systems out of eight were rule-based. In the absence of training examples, handcrafted rules remain the preferred technique for NER (e.g. Sekine and Nobata, 2004 developed a NER system for 200 named entities). In the biomedical domain, rule-based approaches are also used to identify named entities in biomedical literature (see Ananiadou and McNaught, 2006 for more details). The major setback of rule-based approaches is the issue of portability as these approaches are difficult to adapt to different domains. There are three machine learning approaches to build NER systems.

- Supervised Learning Approach
- Semi-Supervised Learning Approach
- Unsupervised Learning Approach

2.3.1 Supervised Learning Approach

Supervised learning is the most dominant technique employed to solve the problem of NER. Supervised learning approach studies the features of positive and negative examples of Named Entities (NEs) over a large collection of annotated documents and learns rules that capture instances of a given type.

Supervised learning techniques include Hidden Markov Models (HMMs) (Bikel et al., 1998; Borkar et al., 2001; Agichtein and Ganti, 2004; Finkel et al., 2005), Decision Trees (Sekine, 1998), Maximum Entropy Models (ME) (Borthwick et al., 1998; Chieu and Ng, 2003; Florian et al., 2007), Maximum Entropy Morkov Models (MEMMs) (McCallum et al., 2000), Support Vector Machines (SVM) (Asahara and Matsumoto, 2003; Mayfield et al., 2003), boosting (Carreras et al., 2003), memory-based learning (MBL) (Meulder and Daelemans, 2003) and Conditional Random Fields (CRF) (McCallum and Li, 2003). All the abovementioned techniques usually consist of a system which reads a large annotated corpus, memorises lists of entities and creates disambiguation rules based on discriminative features. CRFs (McCallum and Li, 2003) are considered as the state-of-the-art method for label assignment to token sequences (words) as it has a more flexible and dominant mechanism for exploiting

[13] http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_toc.html#named

arbitrary feature sets along with dependency in the labels of neighbouring words (Sarawagi, 2008). Apart from IE, supervised learning approaches are frequently used by many other fields of NLP (e.g. Mehdi et al., 2010 used the supervised learning approach for the summarisation of legal documents).

The major shortcoming of the supervised learning approach is the requirement of a large annotated corpus which is sometimes difficult to obtain.

2.3.2 Semi-supervised Learning Approach

As mentioned earlier, semi-supervised approaches rely on the process of bootstrapping. There are many systems which have used this bootstrapping technique for NER.

Brin (1998) used regular expressions in order to generate lists of book titles paired with book authors from the Web. The system started with a few seed examples and learned new ones. The main idea of this algorithm is that many websites conform to a reasonably uniform format across the site.

Collins and Singer (1999) used a parsing technique to search for NE patterns. A pattern is a proper name followed by a noun phrase in apposition. In this system, patterns are kept in pairs {spelling, context} where spelling refers to the proper name and context refers to the noun phrase in its context. The system starts with an initial seed of spelling rules and a candidate which satisfies a spelling rule and they are classified according to how their contexts are accumulated. The most frequent contexts are then turned into a set of contextual rules and later on these rules are used to find further spelling rules and so on. Riloff and Jones (1999) introduced manual bootstrapping technique which consists of a set of entities and a set of contexts. They found out in their experiments that performance of their algorithm deteriorates with the introduction of noise. Cucchiarelli and Velardi (2001) presented a NER system based on Riloff and Jones (1999) manual bootstrapping that used syntactic relations (e.g. subject-object) to discover contextual evidence around named entities.

Pasca et al. (2006) presented a semi-supervised approach for NER by employing Lin’s (1998) distributional similarity measure to generate synonyms (e.g. words which are the members of the same semantic class) for pattern generalisation. They conducted their experiments on a huge corpus (100 million web documents) starting with only 10 seed examples and demonstrated that it is possible to generate one million named entities with a precision of about 88%.

Data selection also plays an important role in the learning process. Heng and Grishman (2006) noted that selection of documents using information retrieval-like relevance measures brought out the best results in their experiments rather than relying on a huge collection of documents.

2.3.3 Unsupervised Learning Approach

Clustering is a typical approach used for unsupervised learning. This approach relies on lexical resources (WordNet), lexical patterns and statistics computed on a large unannotated corpus.

Alfonseca and Manandhar (2002) presented an approach to address the problem of assigning a label to an input word with the appropriate NE type. They made use of WordNet synset and the surrounding context of an input word. Evans (2003) presented an NER system based on the idea of hypernyms described by Hearst (1992) in order to identify named entities. Shinyama and Sekine (2004) presented an approach based on an observation that NEs often appear synchronously in several news articles, whereas common nouns do not. This approach allows identification of rare NEs in an unsupervised manner and can be useful in combination with other NER methods.

Nadeau et al. (2006) presented an unsupervised approach for NER. Their approach made use of simple heuristics based on the work of Mikheev (1999), Petasis et al. (2001) and Palmer and Day (1997) to perform NE disambiguation. Their approach can be divided into two stages. In the first stage, a large gazetteer of entities (list of entities) was created and in the second stage heuristics were used to identify and classify NEs in the given context of a document. They evaluated their system performance against the basic supervised system using the MUC-7 NER corpus (Chinchor, 1998). The supervised system was able to achieve high precision but low recall while the unsupervised system achieved higher recall at the cost of lower precision.

Semi-supervised and unsupervised approaches are useful when a large amount of training data is unavailable or difficult to obtain. There is a lot of research being done in the area of NER spreading across various languages, domains and textual genres (Nadeau and Sekine, 2007). A supervised learning approach gives good performance in the presence of huge collections of annotated data while semi-supervised and unsupervised approaches promise fast deployment of many NE types without the prerequisite of an annotated corpus (Nadeau and Sekine, 2007).

2.4 Rule-based Approaches to Relation Extraction

Relation Extraction (RE) is the second most integral part of any IE system after the NE extraction task. Most of the rule-based approaches in IE rely on hand-written rules or dictionaries and do not learn from annotated examples. In this section we review a few of the well-known rule-based approaches employed in relation extraction.

2.4.1 AutoSlog

Riloff (1993) presented a system called AutoSlog to handle the bottleneck of knowledge engineering. AutoSlog is based on the idea of automatically constructing a "concept dictionary" for an information extraction task. The AutoSlog approach is based on the selective concept extraction method. Selective concept extraction is a form of extraction that selectively processes relevant texts while effectively ignoring irrelevant texts. CIRCUS proposed by Lehnert (1990) is employed for shallow sentence analysing. In order to extract information from texts CIRCUS depends on concept nodes. Concept nodes are an integral part of the AutoSlog system. A concept node consists of a triggering lexical item, enabling conditions in the context and case frame. The AutoSlog algorithm employed a set of heuristics to determine which words and phrases are more likely to activate useful concept nodes and assumes that the verb will determine the role of noun phrase (NP). The AutoSlog system requires human intervention in order to filter out bad concept node definitions wrongly introduce by heuristics or shallow parser failures. A dictionary for the domain of terrorist events (MUC-4) was constructed in only 5 person-hours using AutoSlog. AutoSlog was evaluated against a manually built dictionary which required approximately 1500 person-hours effort and achieved 98% of the performance of manually built dictionary.

2.4.2 PALKA

PALKA (Parallel Automatic Linguistic Knowledge Acquisition) system, presented by Kim J-T and Moldovan (1995), uses knowledge-based information from text for the automatic acquisition of linguistic patterns. PALKA uses an induction method to produce the extraction rules as a pair of a meaning frame and a phrasal pattern, called Frame-Phrasal pattern structure (FP-structure). Patterns are constructed using this FP-structure from training texts and the acquired patterns are then generalised using inductive learning mechanism. PALKA creates a new rule if existing rules cannot be used and then generalises it with the existing ones to include a new positive instance.

In the next Sections (2.5 – 2.7), we will look at various machine learning approaches to relation extraction. A good overview of the machine learning approaches for relation extraction is provided by McDonald, 2005 and Bach & Badaskar, 2007.

2.5 Supervised Approaches to Relation Extraction

The supervised approaches for relation extraction rely on user involvement to provide training examples for the learning process. Supervised approaches rely on training data to induce extraction rules. This section critically reviews supervised approaches to relation extraction. These systems use rule learning algorithms to automatically generate relation extraction patterns from annotated text corpora.

2.5.1 CRYSTAL

Soderland et al. (1995) presented a system called CRYSTAL based on the concept of automatic creation of dictionaries to identify relevant information from a training corpus. The CRYSTAL system takes texts which have been processed by a syntactic parser. A domain expert is required to automatically annotate training documents. From these training documents CRYSTAL learns extraction rules. Inductive learning is used to find similar rules and merges them together by finding the most restrictive constraints that cover both rules.

2.5.2 LIEP

Huffman (1996) presented the LIEP system which learns dictionaries of extraction patterns directly from user-provided examples of texts and events to be extracted from them. The LIEP system uses multi-slot rules for extraction; it lets the user identify events of interest in texts as the system is based on the assumption that an automated training corpus is difficult to obtain. The LIEP system tries to choose extraction patterns which will maximize the positive examples. If a new example cannot be matched by a known pattern, LIEP attempts to generalize a known pattern to cover the example. If generalization is not possible a new pattern is constructed.

2.5.3 WHISK

Soderland (1999) presented the supervised learning system known as WHISK. WHISK uses a machine learning algorithm to deduce regular expressions that are later used as extraction rules. A user annotates the events presented in a set of sentences and WHISK then learns rules from these examples. WHISK has two pre-processing stages: semantic classes in which named entities are marked and chunking parse in which each sentence broken down into groups of words. WHISK annotates more sentences and the rules which disagree with the new examples are rejected. Rules are learned for each sentence not covered by the existing rules and this process continues until all sentences are covered.

2.5.4 GATE

Cunningham et al. (2002) presented GATE (General Architecture for Text Engineering), a graphical development environment enabling researchers/users to develop and deploy various language engineering components and resources. It contains many useful tools that can be used individually or together with other tools.

ANNIE, A Nearly-New IE system is one of them. ANNIE contains a tokeniser, a sentence splitter, a PoS tagger, a gazetteer, a finite state transducer, an orthomatcher and a coreferencer. In the first step, the tokeniser splits text into tokens (e.g. words, punctuations etc). The sentence splitter then segments these tokens into sentences. The PoS tagger is used to annotate these tokens with their PoS tags. The gazetteer consists of a list of named entities (e.g. lists of cities, organisations etc). A finite state transducer/ semantic tagger contains handcrafted rules that illustrate patterns to match and as a result annotation to be created. The orthomatcher recognises relations between named entities and the coreferencer finds identity relations between named entities in the text.

GATE is quite user-friendly and has an easy-to-use environment which provides extensive facilities to researchers for annotation. The annotation can be done manually or semi-automatically by running some processing resources over the corpus. GATE was first implemented as a rule-based system and later on it was supplied with the functionality to perform IE using supervised machine learning. GATE has provided a number of useful facilities to researchers to address various ranges of issues in the area of NLP application development. It is quite robust and scalable.

2.6 Semi-supervised Approaches to Relation Extraction

In this section we critically review the semi-supervised approaches to relation extraction proposed so far.

2.6.1 AutoSlog-TS

Riloff (1996) presented an improved version of the AutoSlog system known as AutoSlog-TS. Experiments were conducted in three domains terrorist events (MUC-4), joint ventures and microelectronics (MUC-5) and the results were compared against AutoSlog system. One of the drawbacks of the AutoSlog system is that it required an annotated corpus which is quite time-consuming and requires a huge amount of effort. The main idea presented in this paper is that domain-specific expressions will appear more often in relevant documents than in irrelevant ones. The AutoSlog-TS does not require any annotated corpora it only needs a classified corpus: relevant vs. non-relevant. The AutoSlog-TS applies exhaustive processing, after the partial parse it generates an extraction pattern for every noun phrase in the training corpus. This result in a large number of patterns being generated which are then evaluated on the basis of co-occurrence statistics with relevant sub-corpora. The user is involved in the process of judging the patterns' relevance and patterns with a relevance score of (p) < 0.5 are discarded. The experiments were conducted in all three domains. MUC-4 data consisted of 1500 documents (772 relevant); AutoSlog generated 1237 patterns which were manually filtered to 450 in 5 hours while AutoSlog-TS generated 32,345 patterns and after filtering 11,225 relevant patterns were retained. The results of MUC-4 were compared against the results of AutoSlog and it showed that AutoSlog got higher recall while AutoSlog-TS were able to achieve higher precision. Portability is a big issue in a knowledge-based natural language processing system. The AutoSlog-TS reduces user involvement in porting IE systems to a new domain. A human need to provide texts classified as relevant and non-relevant, judge the resulting ranked list of patterns and label the resulting patterns in order to specify which kinds of event they will generate.

2.6.2 Snowball: Extracting Relations from Large Plain-Text Collections

Agichtein and Gravano (2000) presented a semi-supervised relation extraction system known as Snowball system. It was based on the Dual Iterative Pattern Expansion

(DIPRE) algorithm (Brin, 1998). The Snowball system relied on a small set of seed examples and a general regular expression that the named entities must match to generate patterns from the text. Snowball system patterns include named entity tags (e.g. <LOCATION>-based <ORGANISATION>) as compared to DIPRE (e.g. <STRING1>-based <STRING2>). In the Snowball system patterns were generated by clustering similar patterns to the seed examples by using a simple single-pass clustering algorithm. The pattern and tuple evaluation was an integral part of the Snowball system and it kept only those patterns and tuples with a high confidence score. The confidence score of a pattern would be high if it was generated by several highly selective patterns. The Snowball system used a newswire corpus in its experiments; the training collection consisted of 178,000 documents, while the test collection consisted of 142,000 documents. The Snowball system was able to achieved higher precision and recall scores compared to DIPRE. Portability is one of the major advantages of the Snowball system as it requires only a handful of seed examples for each new scenario.

2.6.3 Dependency Tree based Pattern Models

In this section, we will discuss various dependency tree based pattern models for relation extraction and a comparison among them.

2.6.3.1 SVO Model

Yangarber et al. (2000) presented the SVO (Subject-Verb-Object) model. The motive behind the approach presented in this paper is to minimise manual labour required in order to construct pattern bases of new domains by using unannotated text, unclassified text and unsupervised learning. The system learns extraction patterns by using dependency parsing and pattern evaluation scores. Patterns used are tuples consisting of four elements: subject, verb, object and phrase referring to either subject or object. According to the presented approach, good patterns are strong indicators of relevant documents. The system starts with a large corpus of documents and a set of useful extraction patterns named as seeds. These patterns are then used to divide the corpus into relevant and irrelevant documents. Relevant documents are those matched by one or more patterns while irrelevant documents are those not matched by any patterns. The patterns which occur more frequently in the relevant documents are selected and added into seeds. The patterns which matched the seed pattern are given the score of 1 and all others 0. The following formula is used to compute the score of each candidate pattern:

$$score\ (p) = \frac{|H \cap R|}{|H|}\log(|H \cap R|)$$

Here H is the set of documents matched by the pattern p and R represents the set of relevant documents. Using the abovementioned formula, the highest scoring pattern is added to the set of accepted patterns. The corpus is first pre-processed to identify named entities and then the Connexor[14] parser is employed for parsing. MUC-6 management succession tasks are used to test the system using the following seed patterns:

```
COMPANY-{appoint, elect, promote, name}-PERSON
PERSON-{resign, depart, quit, step-down}
```

The patterns produced by the system cannot be used directly for extraction so it is difficult to apply the MUC-6 approach for evaluation. Evaluation is therefore based on how accurately patterns match relevant documents and do not match irrelevant ones. A corpus consisting of 100 MUC-6 test documents and 150 documents randomly chosen from the main corpus was used for this purpose.

The main advantage of the presented system is that it offers an unsupervised approach without any need of annotated examples. The disadvantage of this approach is that patterns can not be used directly for a RE task so it can only be evaluated on a text filtering task rather than extraction.

[14] www.connexor.com/

2.6.3.2 Chain Model

Sudo et al. (2001) presented a tree-based pattern representation approach where a pattern is represented as a path in the dependency tree of a sentence. Previous approaches described in Riloff (1996) and Yangarber et al. (2000) are based on one common assumption that relevant documents contain good patterns. Both approaches rely on the sentence structure of English. These approaches failed in case of free word order languages like Japanese. This paper offers an alternative approach for the automatic acquisition of patterns. In the first stage, a morphological analyser and NE-tagger are employed to do document pre-processing. The second stage retrieves the relevant document set from which the relevant sentence set is extracted. Finally all the sentences in the relevant sentence set are parsed and the system takes those paths with frequency higher than a certain threshold as extracted patterns.

Mainichi-Newspaper-95 and Mainichi-Newpaper-94 corpora are used for training and testing the system respectively. The system achieves quite a low recall; moreover this pattern representation may not be able to adequately represent pattern context either.

2.6.3.3 Subtree Model

Sudo et al. (2003) describes the limitations of the previous two extraction pattern models (Yangarber et al., 2000 and Sudo et al., 2001) and presents a new subtree model based on subtrees of the dependency tree. The evaluation shows that the proposed model outperforms the previous models. The SVO model (Yangarber et al., 2000) is based upon the direct syntactic relation between a predicate and its arguments. This pattern representation model is limited in what it can extract from a sentence. The chain model (Sudo et al., 2001) pattern representation may not be able to represent the context of a pattern adequately. The subtree model is the generalisation of the two abovementioned pattern models. According to this model any subtree of a dependency tree can be regarded as an extraction pattern candidate and so it contains all of the patterns proposed by the previous two models. The experiments are conducted using two sets of Japanese texts: Management succession scenario and Murder/Arrest scenario. The process of obtaining extraction patterns consists of following three stages: pre-processing, document retrieval and ranking candidate patterns. Patterns for each model are generated and ranked. The following formula is used for the ranking of subtree patterns.

$$score_i = tf_i \left(\log \frac{N}{df_i} \right)^{\beta}$$

Where:

- tfi – the frequency of pattern *i* in relevant documents
- dfi – the number of docs containing pattern *i*
- N – total number of document in the collection
- β – used to control weight on the *dfi* portion

The advantages of the subtree model are that it allows the capture of more varied context and can extract more specific scenario patterns while the disadvantage of this approach is that it adds the additional complexity of processing a large number of patterns.

2.6.3.4 Linked Chain Model

Greenwood et al. (2005) presented a novel approach which makes use of more complex pattern models than previous approaches. The approach presented a new pattern model 'Linked Chain Model' which is the extension of chain models (Sudo et al., 2003). It joins the pairs of chains which share a common verb root but no direct descendants. The motivation behind this approach is that language is frequently used to articulate the same information in different ways. So this approach learns patterns automatically by identifying patterns with similar meanings to a set of seed patterns.

In order to extract patterns from the corpora, the paper uses a weakly supervised bootstrapping method similar to Yangarber (2003) which learns patterns from a corpus based upon their similarity to seed patterns. The paper ranked learned patterns by employing an iterative algorithm which compares each candidate pattern against the centroid vector of the currently accepted patterns. The four highest scoring patterns in each iteration are then added to the accepted patterns.

2.6.3.5 A Semantic Approach to IE Pattern Induction

Stevenson and Greenwood (2005) presented an alternative approach to Yangarber et al. (2000) for learning IE patterns. The approach is based on the assumption that patterns with similar meanings are expected to be valuable for extraction. The algorithm presented in this paper shows that this approach performs well when compared with the previously reported document-centric approach. The approach uses iterative learning algorithm for pattern learning, which starts with a set of seed patterns which are identified to be useful extraction patterns and compares every other pattern with the ones acknowledged to be good and then selects the highest scoring of these and adds them to the set of good patterns. This process continues until enough patterns have been learned. The approach is evaluated using two evaluation regimes: document filtering and sentence filtering

In document filtering the task involves identifying relevant documents from irrelevant ones while sentence filtering evaluates how accurately generated patterns can distinguish between relevant and non-relevant sentences. The results produced by this approach are much superior to those produced by Yangarber et al. (2000). This approach failed to represent events which cannot be described as SVO structure so a more expressive model is required.

2.6.3.6 Comparing IE Models

Stevenson and Greenwood (2006) compared the four previously reported pattern models based on dependency trees and evaluated them using three different dependency parsers. The results of the experiments conducted in this paper show that linked chain pattern models perform better than the other models. The choice of a pattern model is very important for any extraction task. The pattern model should be expressive enough to extract the required information from a parse of a dependency tree accurately. SVO model (see Section 2.6.3.1) used subject-verb-object tuples from the dependency tree as extraction patterns. The SVO model is unable to represent linguistic constructions such as nominalisations and prepositional phrases. Chain model (see Section 2.6.3.2) has the ability to represent the information expressed as a nominalisation or prepositional phrase but this model is unable to represent sentences containing transitive verbs and it also fails to represent the context of a pattern adequately. The linked chain model (see Section 2.6.3.4) is able to encode the information represented by both SVO and chain models collectively. The subtree model (see Section 2.6.3.3) is richer in terms of information representation as compared to the abovementioned models but it produces too many patterns which are an uphill task to compute and so it adds additional complexity. The experiments are conducted on newspaper texts and biomedical texts using three dependency parsers in order to find suitable pattern representation models for encoding the information of interest to IE systems. Three dependency parsers used in these experiments are: MINIPAR[15] (Lin, 1999), the Machinese Syntax[16] parser (Tapanainen and Järvinen, 1997) and the Stanford[17] parser (Klein and Manning, 2003). SVO model and chain model performed poorly and provided less coverage while the linked chain models achieved a bounded coverage of 95% which means that this model can represent the majority of relations present in the dependency tree.

Stevenson and Greenwood (2009) presented an analysis of various models' performance on two different textual domains: management succession and biomedical text. Their analysis reveals that there is a wide variation between the models' performance. In this paper, each pattern model was analysed in terms of its ability to represent relevant information, number of generated patterns and performance on an IE scenario. The experiments result showed that the linked chain model performance is quite promising compared to other pattern models.

2.7 Unsupervised Approaches to Relation Extraction

In this section, we will review a few of the most recent unsupervised approaches to relation extraction.

Hasegawa et al. (2004) presented an unsupervised approach for the discovery of relations among named entities from a newspaper domain. Their approach employed

[15] http://webdocs.cs.ualberta.ca/~lindek/minipar.htm
[16] www.connexor.com/software/syntax/
[17] http://www-nlp.stanford.edu/software/lex-parser.shtml

the clustering technique in order to cluster named entity pairs according to the similarity of context words intervening between them. The relation discovery process was based on the assumption that pairs of named entities co-occurring in similar context can be grouped together in a cluster. After the NER, the two named entities are considered to co-occur if they appear within the same sentence and are separated by at most *N* intervening words. A vector space model and cosine similarity measures were employed to calculate the similarities between the set of contexts of named entities pairs. The approach used the maximum 5 context words between named entities and set the frequency threshold of 30 co-occurring named entities pairs. The presented approach was able to achieve a good precision and recall but one of the drawbacks of this approach is that because of high frequency threshold, the system was unable to discover some valuable relations.

Sekine (2006) and Shinyama and Sekine (2006) presented two unsupervised approaches to IE known as ‘On-demand IE’ and ‘Pre-emptive IE’ respectively. The basic motive behind both these approaches was to identify the most salient relations in documents and extract information on user demands by employing unsupervised learning methods. The on-demand IE system (Sekine, 2006) extracts salient relations from the text based on a user query and builds tables based on these extracted relations by using paraphrase discovery technology. The system makes use of recent advances in pattern discovery, paraphrase discovery and extended NE tagging. The system used a newspaper corpus and retrieves relevant documents based on a user query and then applies PoS tagger, a dependency analyser and an extended NE tagger to extract patterns from the relevant documents. These extracted patterns are then arranged into a set of similar patterns by applying paraphrase recognition. A table was created for each pattern set, if the pattern set contained more than two patterns. Shinyama and Sekine (2006) (pre-emptive IE) apply NER, coreference resolution and parsing to a newspaper corpus in order to extract relations between NEs. The approach uses unrestricted relation discovery in order to discover all possible relations from texts and presents them as tables. In unrestricted relation discovery the relations appearing repeatedly in a corpus are extracted automatically (without human intervention). The extracted relations are grouped into pattern tables of NE pairs expressing the same relation. This approach uses clustering in order to cluster the semantically similar relations.

Etzioni et al. (2008) presented an unsupervised approach to RE by using Web as a corpus. Their approach used a huge corpus of 9 million web pages to automatically extract all relations between noun phrases. The main contribution of this approach is to introduce an open RE system known as *TEXTRUNNER*. TEXTRUNNER consists of three key modules: self-supervised learner, single-pass extractor and redundancy-based assessor. Self-supervised learner module produces a classifier by using a small sample corpus without any hand-tagged data. This classifier labels candidate extractions as 'trustworthy' or not. The single-pass extractor module makes a single pass over the whole corpus to extract tuples of all possible relations from corpus. These extracted tuples are then sent to the classifier and only those which the classifier labels as trustworthy are kept. A redundancy-based assessor module assigns a probability score to each trustworthy tuple based on a probabilistic model of redundancy in text (Downey et al., 2005). The experimental results revealed in this paper show that TEXTRUNNER achieves a 33% relative error reduction for a comparable number of extractions when compared with the state-of-the-art Web RE system KNOWITALL (Etzioni et al., 2005). Moreover, TEXTRUNNER was able to achieve higher precision than KNOWITALL.

Eichler et al. (2008) presented an unsupervised RE system (IDEX) which automatically extracts information regarding an input topic provided by the user. The relevant documents related to the given topic are then retrieved and extracted relations are clustered in an unsupervised way. IDEX employs LingPipe[18] for sentence boundary detection, NER and coreference resolution. IDEX only considered those sentences for relation extractions which contain at least two NE's. These selected sentences are then parsed using Stanford parser[19]. IDEX then extracts all the verb relations i.e. for each verb its subject(s), object(s), preposition(s) with arguments and auxiliary verb(s) and it keeps only those verb relations where at least the subject or object is an NE. Extracted relations are grouped into relation clusters based on their similarity. IDEX used Berlin Central Station corpus for their experiments which comprise 1068 web pages downloaded from Google consisting of 55255 sentences, 10773 relation instances were automatically extracted and clustered by those

[18] http://alias-i.com/lingpipe/

[19] http://nlp.stanford.edu/

sentences. The system was able to produce 306 clusters out of which 121 were deemed as consistent (i.e. all instances in the cluster express similar relations), 35 partly consistent and 69 were not consistent.

2.8 Relation Extraction in the Biomedical Domain

There is a large body of research dedicated to the problem of extracting relations from general-domain texts, and from biomedical texts in particular. BioNLP[20] has played a great role in biomedical research by providing a platform with useful resources to the research community. Most previous approaches have been supervised and tried both to extract relations and assign labels describing the semantic types of the relations (Cunningham et al., 2002; Zelenko et al., 2003; Culotta and Sorensen, 2004; Bunescu and Mooney, 2006 among many others). These approaches required a manually annotated corpus, which is very laborious and time-consuming to produce (see Section 2.5).

Semi-supervised and unsupervised approaches rely on seed patterns and/or examples of specific types of relations (see Section 2.6 and Section 2.7). As is known from literature, RE in the biomedical domain is quite difficult as compared to other domains, such as the news domain, due to the inherently complex nature of text in the biomedical domain (e.g. Cohen and Hersh, 2005). As sentences in the biomedical domain are syntactically complex, the subsequent RE phase depends upon the correct identification of the NEs and correct analysis of linguistic constructions expressing relations between them. In the biomedical domain, most work has focused on fully supervised or semi-supervised approaches. For example, Wong (2001) used templates to determine protein-protein interactions from biomedical text. Most of the supervised approaches relied on regular expressions to learn patterns, while semi-supervised approaches exploited pre-defined seed patterns and cue words (Ananiadou and McNaught, 2006).

[20] http://www.bionlp.org/

Blaschke et al. (1999) presented a system for the automatic detection of protein-protein interactions from the scientific abstracts. Their approach relies on pre-specified protein names and a set of verbs that represent the actions. This paper does not provide any precision or recall scores.

Ono et al. (2001) presented a system for the extraction of protein-protein interactions from biomedical literature. The system employed certain sets of regular expression rules and cue words (“interact”, “bind”, etc.) along with a protein name dictionary to extract the relation between two proteins. The system achieved a high performance with a precision rate of 94% and a recall rate of 85%. One of the shortcomings of this approach is its inability to deal with the complex sentences that distance a subject or object from a verb.

Huang et al. (2004) presented a data-driven approach for the extraction of protein-protein interactions from biomedical literature. Their approach employed a dynamic programming algorithm along with a protein dictionary in order to compute distinguishing patterns by aligning relevant sentences and key verbs that describe protein-protein interactions. Their system was able to attain a precision of 80.5% and a recall of 80%.

Corney et al. (2004) describes a system known as BioRAT which constructs templates using a set of regular expressions, part-of-speech, gazetteer categories, literal strings and words. BioRAT is designed to give the people a powerful tool in order to locate and analyse research papers. BioRAT plays the role of a research assistant by finding relevant documents relevant to a given query and automatically highlighting the salient facts in each document. BioRAT was able to achieve a precision of 55.7% and a recall of 20.3% on biomedical abstracts and precision and recall scores of 51.25% and 43.6% respectively on full-length papers.

Martin et al. (2004) presented another approach based on pattern matching, the approach extracted protein-protein interactions using a number of dictionaries containing: protein names and their synonyms, protein interaction verbs and their synonyms and common strings used which are helpful in the identification of unknown proteins.

Fundel et al. (2007) developed a tool known as RelEx to extract biomedical relations (protein-gene interactions) from free text in a biomedical literature. This tool was based on dependency trees along with rules to process these trees. For NER of gene and protein names this tool employed a synonym dictionary (Fundel and Zimmer, 2006) while a list of restriction terms was used to specify relations of interest in the text. RelEx extracted relations from dependency trees by extracting paths connecting pairs of proteins while making sure that these paths contain relevant terms describing the relation between the given pair of proteins. RelEx was evaluated using a comprehensive set of one million MEDLINE abstracts dealing with gene and protein relations and was able to attain 80% precision and 80% recall.

All of the aforementioned approaches mostly rely on pattern matching and require a large number of patterns in order to extract the desired information. Overall, there has been little work on fully unsupervised approaches to RE, ones that would be able to locate significant relations in a particular collection of texts. Semi-supervised approaches, while offering considerable savings on the preparation of training data, are still limited to pre-defined types of relations that have to be instantiated in either seed extraction patterns, seed pairs of related named entities, or annotated examples. Relation Extraction in the biomedical domain has been addressed primarily with either supervised approaches or those based on manually written extraction rules, which are rather inadequate in scenarios where relation types of interest are not known in advance.

2.9 Use of Web as a corpus

The Web is the largest possible source of free textual data, containing hundreds of billions of words in various languages and consistently growing at a rapid pace. Presently, many researchers use the Web[21] as a data source in their research. The Web enables researchers to handle the data sparseness bottleneck in various NLP applications. Killgarriff and Grefenstette (2003) shed light on the use of the Web as a

[21] http://www.webcorp.org.uk/

corpus for many NLP applications. They argued that having large amounts of data would improve performance more than fine-tuning algorithms. Manning and Schütze (1999) suggested that having a large amount of training data (as in the case with the Web corpus) is very useful for many statistical NLP applications. In many NLP applications the algorithms which used the Web as a corpus were successful at many linguistics tasks and frequently surpassed sophisticated methods based on traditional corpora (e.g. Turney, 2001; Keller and Lapata, 2003).

The Web is a huge source of information and it has a huge impact in the field of NLP but it has its drawbacks, too. One main drawback of using the Web as a corpus is that along with text types it also contains a lot of useless material. Another disadvantage is that it is impossible to replicate an experiment in an exact way at a later time as the Web is constantly in flux and growing at a rapid pace. Apart from redundancy, one of the other main criticisms of using the Web as a corpus is that it is not balanced as an ideal or traditional corpus should be and due to that, the data obtained from the Web corpus might not be representative. On the other hand, Killgarriff and Grefenstette (2003) argued that no corpus is completely balanced and representative.

The Web is also quite frequently used by many researchers in the area of IE. Brin (1998) presented an approach known as Dual Iterative Pattern Relation Extraction (DIPRE) which extracted relations (book titles and authors) from the Web, automatically or with minimal human intervention. Due to the progress made in computer hardware, many IE researchers have used unsupervised approaches based on the Web e.g. Sekine, 2006; Shinyama and Sekine, 2006; Banko et al., 2007 and Eichler et al., 2008 (see Section 2.7 for further details). Mukherjea and Sahay (2006) used the Web in order to automatically discover biomedical relations. Their approach relied on the retrieval of relevant information from web search engines by employing various lexico-syntactic patterns as queries.

In our research, we will carry out our experiments using traditional corpora as well as the corpus collected from the Web and will compare the results obtained from these corpora.

2.10 Summary

In this chapter, we have discussed various approaches presented so far in order to automatically generate multiple choice test items. We also elaborated the concept of IE, its subtasks, its main components, its evaluation and approaches to build IE systems and its applications in a real world. IE has two main components: NER and RE. We have described various supervised, semi-supervised and unsupervised approaches for each component. We also looked at the various dependency tree based patterns models and comparison among these models in this chapter. At the end of this chapter, we also discussed the growing trend of using the Web as a corpus, its advantages and disadvantages.

Chapter 3: Stem Sentences Selection via IE

In this chapter, we will discuss the IE component of our system (see Figure 2 in Section 1.7). We will investigate two unsupervised approaches (surface-based and dependency-based) to Relation Extraction to be applied in the context of automatic generation of multiple-choice questions (MCQs).

Our assumption for Relation Extraction is that it is between Named Entities stated in the same sentence and that presence or absence of a relation is independent of the text prior to or succeeding the sentence. This connotes that only information obtained from sentences including the two Named Entities will be relevant for Relation Extraction.

In the surface-based approach, we will examine three different surface pattern types, each implementing different assumptions about linguistic expression of semantic relations between Named Entities while in the dependency-based approach we will explore how dependency relations based on dependency trees can be helpful in extracting relations between Named Entities. We will evaluate both these approaches in terms of precision, recall and F-score. Our experiments make use of traditional corpora along with the similar corpus collected from the Web. At the end of this chapter, we will perform a comparison between the surface-based approach and the dependency-based approach.

3.1 Unsupervised Surface-based Patterns

The approach aims to identify the most important semantic relations in a document without assigning explicit labels to them in order to ensure broad coverage, unrestricted to predefined types of relations, which is particularly important in the context of testing learners' familiarity with learning material.

Our main findings indicate that the approach is capable of achieving high precision scores and its enhancement with linguistic knowledge helps to produce significantly improved patterns. The intended application for the proposed method is in the context of an e-Learning system for automatic assessment of students' comprehension of training texts; however it can also be applied to other NLP scenarios, where it is necessary to recognise important semantic relations without any prior knowledge as to their types.

Information Extraction (IE) is an important problem in many information access applications. As mentioned in Chapter 2, Named Entity Recognition (NER) and Relation Extraction (RE) are the two integral components of any IE system. The first step is the identification of the NEs present in the text. These NEs will be different depending on the nature of the text and the intended application. Following the identification of NEs the next step is the RE phase. The goal is to identify all the instances of specific semantic relations between NEs of interest in the text. For this purpose RE patterns are used to recognise and label these relations.

3.1.1 Our Approach

The main advantage of our approach (Afzal and Pekar, 2009) is that it can cover a potentially unrestricted range of semantic relations while other supervised and semi-supervised approaches (see Section 2.5 and Section 2.6) can learn to extract only those relations that have been exemplified in annotated text, seed patterns or seed named entities. Moreover, our approach is very suitable for situations where a lot of unannotated text is available as it does not require manually annotated text or seeds. Such an approach can be useful, specifically, in such applications as Multiple-Choice Question generation (Mitkov et al., 2006; see Section 2.1) or a pre-emptive approach in which viable IE patterns are created in advance without human intervention (Shinyama and Sekine, 2006; Sekine, 2006; see Section 2.7). Figure 3 shows the whole architecture of our approach. We elaborate the NER process in Section 3.1.2; Section 3.1.3 explains the process of candidate patterns extraction. Section 3.1.4 describes various information theoretic measures and statistical tests for patterns ranking depending upon patterns associations with a domain corpus while Section

3.1.5 discusses the evaluation procedures and the experimental results are discussed in Section 3.1.6.

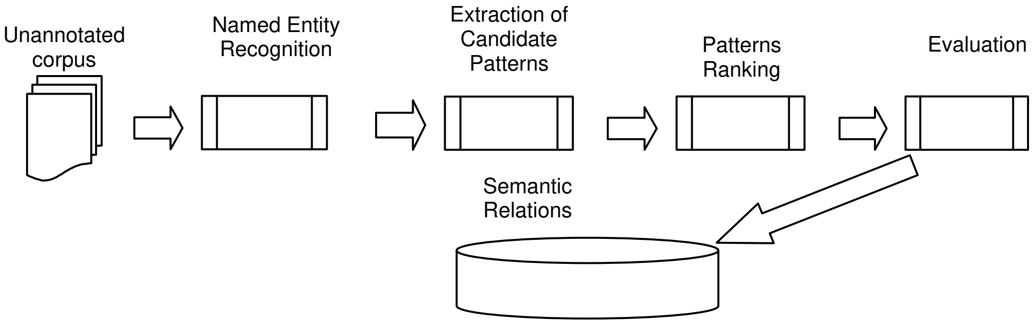


Figure 3: Relation Extraction approach

We will employ this approach for the automatic generation of MCQs, where it will be used to find relations and NEs in educational texts that are important for testing students' familiarity with key facts contained in the texts. In order to achieve this, we need an IE method that has a high precision and at the same time works with unrestricted semantic types of relations (i.e. without reliance on seeds), while recall is of secondary importance to precision.

3.1.2 NER and PoS Tagging of Biomedical Texts

Biomedical NER is generally considered to be more difficult than other domains like newswire text. There is huge number of NEs in the biomedical domain and new ones are constantly added (Wilbur and Smith, 2007) which means that neither dictionaries nor the training data approach will be sufficiently comprehensive for NER. The volume of published biomedical research has expanded at a rapid rate in the recent past. MEDLINE[22] (Medical Literature Analysis and Retrieval System Online) is the U.S. National Library of Medicine containing over 18 million references to journal articles regarding biomedicine. MEDLINE is currently growing at the rate of over 600,000 new citations each year[23]. PubMed[24], a search engine, is used to access the MEDLINE content. NER in the biomedical domain has been researched over the

[22] http://www.nlm.nih.gov/pubs/factsheets/medline.html
[23] http://www.nlm.nih.gov/bsd/stats/cit_added.html
[24] http://www.ncbi.nlm.nih.gov/pubmed/

years with various challenges such as BioCreAtIvE[25] (Critical Assessment of Information Extraction systems in Biology) and shared tasks in conferences addressing the issues and evaluating the performances of various named entity recognition systems.

Named entities (NEs) in the biomedical domain are expressed in various linguistic forms such as abbreviations, plurals, compound, coordination, cascade, acronyms and apposition (Zhou et.al, 2004). These various linguistic forms are exemplified in Table 2 (Ananiadou and McNaught, 2006).

Linguistic Forms	**Example Gene and Protein Names**
Abbreviation	GLA
Plural	p38MPAKs, ERK1/2
Compound	Rpg1p/Tif32p
Coordination	91 and 84 kDa proteins
Cascade	kappa 3 binding factor (such that *kappa 3* is a gene name)
Description	an inhibitor of p53
Acronym	Phospholipase D (PLD)
Apposition	PD98059, specific MEK1/2 inhibitor

Table 2: Example gene and protein names in various linguistic forms

One NE can be used to represent different concepts which results in further ambiguities, for example '*ferritin*' can be a biological substance or a laboratory test. Moreover, many biological NEs have several names e.g. '*PTEN*' and '*MMACI*' refer to the same gene which in turn makes NER in the biomedical domain more difficult. Another problem is that authors frequently do not follow existing naming conventions, instead introducing their own abbreviations and using them throughout the papers (Chen et al., 2005). Moreover, the NEs in the biomedical domain are much longer on average than NEs from other domains. It is generally much easier for both human and automated systems to find out whether an NE is present than to detect its

[25] http://biocreative.sourceforge.net/

boundaries (Yeh et al., 2005) as the case is not always a reliable indicator of sentence boundaries (e.g. a new sentence can start with lowercase word in a biomedical domain). Yeh at al. (2005) also compared the length distribution of gene names with the length distribution of organisation names in the newswire domain. Their results revealed that the average length of gene names was 2.09 compared to 1.69 for organisation names.

Due to the syntactic and semantic complexity of the biomedical domain many IE systems have utilised tools (e.g. part-of-speech tagger, NER, parsers, ontologies) specifically designed and developed for the biomedical domain (e.g. Andrade and Valencia, 1998; Pustejovsky et al., 2001, 2002). Moreover, Grover et al. (2005) presented a report investigating the suitability of current NLP resources for syntactic and semantic analysis for the biomedical domain. The GENIA tagger[26] is a specific tool designed for biomedical texts, which is used to analyse English sentences and outputs the base forms, part-of-speech tags, chunk tags and NE tags. The GENIA part-of-speech tagger is trained on a general domain corpus (Wall Street Journal corpus) as well as GENIA corpus and PennBioIE corpus (Kulick et al., 2004). Due to this the GENIA part-of-speech tagger is able to handle various kinds of biomedical text, and achieves a very high accuracy on biomedical text. Table 3 shows the tagging accuracies of a tagger trained on different data sets (Tsuruoka et al., 2005; Tsuruoka and Tsujii, 2005).

	Wall Street Journal (WSJ) corpus	**GENIA corpus**
A tagger trained on WSJ corpus	97.05%	85.19%
A tagger trained on GENIA corpus	78.57%	98.49%
GENIA tagger	96.94%	98.26%

Table 3: Tagging accuracies

The GENIA tagger produces the output in the following format:

word1 base1 POStag1 chunktag1 NEtag1
word2 base2 POStag2 chunktag2 NEtag2

[26] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

: : : : :

The tagger represents the chunk tags in the IOB format (B for BEGIN, I for INSIDE and O for OUTSIDE). The NE tagger is designed to recognise mainly the following named entities: protein, DNA, RNA, cell_type and cell_line. The NE tagger is trained on the NLPBA data set[27], a shared task of biomedical NE recognition that was held from March to April 2004. The task main objective was to identify and classify terms in bio-molecular biology which correspond to instances of concepts which are of particular interest to biologists. Table 4 shows the performance of GENIA NER[28].

Entity Type	**Precision**	**Recall**	**F-score**
Protein	65.82	81.41	72.79
DNA	65.64	66.76	66.20
RNA	60.45	68.64	64.29
Cell Line	56.12	59.60	57.81
Cell Type	78.51	70.54	74.31
Overall	67.45	75.78	71.37

Table 4: GENIA NER performance

3.1.3 Extraction of Candidate Patterns

Our general approach to the discovery of interesting extraction patterns consists of two main stages: (i) the construction of potential patterns from an unannotated domain corpus and (ii) their relevance ranking.

3.1.3.1 Linguistic types of patterns

Once the training corpus has been tagged with the GENIA tagger, the process of pattern building takes place. Its goal is to identify which NEs are likely to be semantically related to each other.

[27] http://research.nii.ac.jp/~collier/workshops/JNLPBA04st.htm

[28] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

The procedure for constructing candidate patterns is based on the idea that important semantic relations are expressed with the help of recurrent linguistic constructions, and these constructions can be recognised by examining sequences of content words (nouns, verbs, adjectives and adverbs) appearing between NEs. Semantic patterns are widely used in the area of IE. As in IE, we are interested in extraction of semantic classes of objects (NEs), relationships among these NEs and events in which these entities participate. To find such constructions, we impose a limit on the number of content words intervening between the two NEs. We experimented with different thresholds and finally settled on a minimum of one content word and a maximum of three content words to be extracted between two NEs. The reason for introducing this condition is that if there are no content words between two NEs then, although some relation might exist between them, it is likely to be a very abstract grammatical relation. For example, in "X of Y" there is a relation between X and Y, but the phrase does not explicitly express any domain-specific knowledge. On the other hand, if there are too many content words intervening between two NEs, then it is likely they are not related at all. We build patterns using this approach and store each pattern along with its frequency in a database. In extracted patterns, lexical items are represented in lowercase while semantic classes are capitalised. For example in the pattern "*PROTEIN encode PROTEIN*", here *encode* is a lexical item while *PROTEIN* is a semantic class.

In this chapter we describe experiments with different surface pattern types each implementing different assumptions about linguistic expression of semantic relation between named entities without prepositions and with the inclusion of prepositions. In the first phase of experiments we consider the following surface pattern types without prepositions:

- Untagged word patterns
- PoS-tagged word patterns
- Verb-centred patterns

The reason for choosing these different types of surface patterns is that verbs typically express semantic relations between nouns that are used as their arguments. *Untagged word patterns* consist of NEs and their intervening content words. Some examples of the most frequent untagged word patterns from GENIA corpus along with their frequencies are shown in Table 5.

Patterns	**Frequency**
PROTEIN activation PROTEIN	53
DNA contain DNA	46
PROTEIN include PROTEIN	43
PROTEIN bind DNA	39
PROTEIN as well PROTEIN	37
PROTEIN expression PROTEIN	35
PROTEIN activate PROTEIN	32
CELL_TYPE express PROTEIN	31
PROTEIN expression CELL_TYPE	29
PROTEIN induce PROTEIN	29

Table 5: Untagged word patterns along with their frequencies

PoS-tagged word patterns contain the PoS of each content word. Table 6 shows examples of the most frequent PoS-tagged word patterns from the GENIA corpus along with their frequencies.

Patterns	**Frequency**
PROTEIN activation_n PROTEIN	53
DNA contain_v DNA	46
PROTEIN include_v PROTEIN	43
PROTEIN bind_v DNA	39
PROTEIN as_a well_a PROTEIN	37
PROTEIN expression_n PROTEIN	35
PROTEIN activate_v PROTEIN	32
CELL_TYPE express_v PROTEIN	31
PROTEIN expression_v CELL_TYPE	29
PROTEIN induce_v PROTEIN	29

Table 6: PoS-tagged word patterns along with their frequencies

Verb-centred patterns contain patterns where the presence of a verb is compulsory in each pattern. Table 7 shows some of the most frequent verb-centred patterns from the GENIA corpus along with their frequencies. We require the presence of a verb in the verb-based patterns as verbs are the main predicative class of words, expressing specific semantic relations between two named entities.

Patterns	**Frequency**
DNA contain_v DNA	46
PROTEIN include_v PROTEIN	43
PROTEIN bind_v DNA	39
PROTEIN activate_v PROTEIN	32
CELL_TYPE express_v PROTEIN	31
PROTEIN induce_v PROTEIN	29
DNA encode_v PROTEIN	27
CELL_LINE express_v PROTEIN	20
PROTEIN involve_v PROTEIN	18
PROTEIN bind_v PROTEIN	18

Table 7: Verb-centred patterns along with their frequencies

Moreover, in the pattern building phase, the patterns containing the passive form of the verb like:

PROTEIN be_v express_v CELL_TYPE

are converted into the active voice form of the verb like:

CELL_TYPE express_v PROTEIN

Because such patterns were taken to express a similar semantic relation between NEs, passive to active conversion was carried out in order to relieve the problem of data sparseness: it helped to increase the frequency of unique patterns and reduce the total number of patterns. For the same reason, negation expressions (not, does not, etc.)

were also removed from the patterns as they express a semantic relation between NEs equivalent to one expressed in patterns where a negation particle is absent.

In addition, patterns containing only stop-words (a list of English stop-words common in IR) were also filtered out. Table 8 shows a few examples of stop-word patterns which were filtered out during the candidate pattern construction.

DNA through PROTEIN
PROTEIN such as PROTEIN
PROTEIN with PROTEIN in CELL_TYPE
PROTEIN be same in CELL_LINE
PROTEIN against PROTEIN

Table 8: Patterns only containing stop-words

3.1.4 Pattern Ranking

After candidate patterns have been constructed, the next step is to rank the patterns based on their significance in the domain corpus. The ranking method we use requires a general corpus that serves as a source of examples of use of the patterns in domain-independent texts. To extract candidates from the general corpus, we treated every noun as a potential named-entity holder and the candidate construction procedure described above was applied to find potential patterns of the three different types in the general corpus. Some of these ranking methods have been used in classification of words according to their meanings (Pekar et al., 2004) but to our knowledge this approach is the first one to explore these ranking methods to rank IE patterns. We used these ranking methods in our research as they are more appropriate for our unsupervised RE approach as compared to the pattern ranking method used by semi-supervised approaches (Yangarber et al., 2000; Sudo et al., 2001; Sudo et al., 2003), where tf-idf is used in order to iteratively collect IE patterns and relevant documents from a collection of relevant and irrelevant documents.

In order to score candidate patterns for domain-relevance, we measure the strength of association of a pattern with the domain corpus as opposed to the general corpus. The patterns are scored using the following methods for measuring the association between a pattern and the domain corpus:

- Information Gain (IG)
- Information Gain Ratio (IGR)
- Mutual Information (MI)
- Normalised Mutual Information (NMI)
- Log-likelihood (LL)
- Chi-Square (CHI)

These association measures were included in the study as they have different theoretical principles behind them: IG, IGR, MI and NMI are information-theoretic concepts while LL and CHI are statistical tests of association.

Information Gain measures the amount of information obtained about domain specialisation of corpus c, given that pattern p is found in it.

$$IG(p,c) = \sum_{d \in \{c,c'\}} \sum_{g \in \{p,p'\}} P(g,d) \log \frac{P(g,d)}{P(g)P(d)}$$

where *p* is a candidate pattern, *c* – the domain corpus, *p'* – a pattern other than *p*, *c'* – the general corpus, *P*(*c*) – the probability of *c* in the "overall" corpus {*c*, *c'*}, and *P*(*p*) – the probability of *p* in the overall corpus.

Information Gain Ratio aims to overcome one disadvantage of IG consisting in the fact that IG grows not only with the increase of dependence between *p* and *c*, but also with the increase of the entropy of *p*. IGR removes this factor by normalising IG by the entropy of the corpus:

$$IGR(p,c) = \frac{IG(g,c)}{-\sum_{g \in \{p,p'\}} P(g) \log P(g)}$$

Pointwise Mutual information has been traditionally used in statistical NLP to measure the association between two linguistic phenomena, such as the elements of a multiword unit. Pointwise MI between corpus *c* and pattern *p* measures how much information the presence of *p* contains about *c*, and vice versa:

$$MI\ (p,c) = \log \frac{P(p,c)}{P(p)P(c)}$$

Mutual Information has a well known problem of being biased towards infrequent events. To tackle this problem, we normalised the MI score by a discounting factor, following the formula proposed in Lin and Pantel (2002).

Chi-Square and Log-likelihood are statistical tests which work with frequencies and rank-order scales, both calculated from a contingency table with observed and expected frequency of occurrence of a pattern in the domain corpus. **Chi-Square** is calculated as follows:

$$x^2(p,c) = \sum_{d \in \{c,c'\}} \frac{(O_d - E_d)^2}{E_d}$$

where *O* is the observed frequency of *p* in domain and general corpus respectively and *E* is the expected frequency of *p* in two corpora. *E* is calculated as:

$$E_i = \frac{N_i \sum_i O_i}{\sum_i N_i}$$

Here N_i is the total frequency of a pattern in corpus *i*.

Log-likelihood is calculated according to following formula:

$$-2\ln\lambda = 2\sum_{i} O_{i}\ln\left[\frac{O_{i}}{E_{i}}\right]$$

This equates to calculating LL as follows:

$$LL(p,c) = 2\left(O_1\log\left(\frac{O_1}{E_1}\right) + O_2\log\left(\frac{O_2}{E_2}\right)\right)$$

where O_1 and O_2 are observed frequencies of a pattern p in the domain and general corpus respectively, while E_1 and E_2 are its expected frequency values in the two corpora.

In addition to these six measures, we introduce a **meta-ranking** method that combines the scores produced by several individual association measures (apart from MI), in order to leverage agreement between different association measures and downplay idiosyncrasies of individual ones. We excluded MI here because of its bias towards infrequent events as mentioned earlier (Lin and Pantel, 2002). Because the association functions range over different values (for example, IGR ranges between 0 and 1), we first normalise the scores assigned by each method:

$$S_{norm}(p) = \frac{s(p)}{\max_{q\in P}(s(q))}$$

where $s(p)$ is the non-normalised score for pattern p, from the candidate pattern set P. The normalised scores are then averaged across different methods and used to produce a meta-ranking of the candidate patterns.

Apart from the aforementioned pattern ranking methods, we also used most frequently used pattern ranking method: tf-idf as a baseline in our experiments too. The tf-idf scoring is commonly used in IR (Manning and Schütze, 1999). Sudo et al (2003) (see Section 2.6.3.3) used this method to rank IE patterns. We used the following formula to rank IE patterns:

$$score_i = tf_i \left(\log \frac{N}{df_i} \right)$$

where tf_i is the frequency of pattern i in domain corpus, df_i the number of documents containing pattern i and N is the total number of documents in the collection (both domain and general corpus).

Given the ranking of candidate patterns produced by a scoring method, a certain number of highest-ranking patterns can be selected for evaluation. We studied two different ways to select these patterns: (i) one based on setting a threshold on the association score below, in which the candidate patterns are discarded (henceforth, *score-thresholding measure*) and (ii) one that select a fixed number of top-ranking patterns (henceforth, *rank-thresholding measure*). During the evaluation, we experimented with different rank- and score thresholding values.

3.1.5 Evaluation

3.1.5.1 Experimental data

We used the GENIA Corpus as the domain corpus while British National Corpus (BNC) was used as a general corpus. The GENIA corpus consists of 2000 abstracts extracted from the MEDLINE containing 18,421 sentences. In the evaluation phase, GENIA EVENT Annotation corpus[29] is used (Kim et.al, 2008). It consists of 1000 MEDLINE abstracts similar to the GENIA corpus and has 9,372 sentences. The main difference between the GENIA and GENIA EVENT corpora is that in the GENIA EVENT corpus events are identified and annotated.

In order to handle the problem of data sparseness due to the small size of the GENIA corpus we developed a WEB corpus (consisting of 132,582 sentences) by collecting MEDLINE articles similar to the GENIA corpus from the National Library of

[29] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi?page=Event+Annotation

Medicine[30]. The Web corpus was collected using a commercial web crawler, which implements a methodology for collecting a topical corpus, similar to the one implemented in tools such as BootCat.[31] The commercial web crawler was preferred over BootCat because it has a term extractor integrated with it, so high quality terms were automatically extracted from pages being analysed and used for automatically building more queries while BootCat extracts single words. It is fully automated, i.e. one does not have to do manual revision of the extracted terms after every iteration. Moreover, it queries multiple search engines (Google, Yahoo and Bing) and so the crawling results are not biased towards any particular search engine. As the commercial web crawler uses a term extractor, it is better at crawling highly technical domains which are best captured by multi-word terms. BootCat, instead, was primarily intended to collect language-specific, topic-independent corpora, where single words are more suitable for collecting content. In response to an original set of manually constructed queries built from the GENIA corpus, original queries were constructed by manually defining several topical terms (named entities) e.g. protein, DNA and combining them randomly to create an initial set of queries. The crawler collects web pages by making calls to several popular search engines, extracts topical terminology from the pages, selects the most promising topical terms to create new queries, and uses them to collect more web pages on the topic. The crawler collected web pages in this iterative manner until the desired size of the corpus is reached. The crawler strips off boilerplate content (navigation menus, standard notices etc.) from each page, removes HTML tags, detects and discards duplicate pages. The GENIA named entity tagger was then used for NER and PoS tagging. The quality of the collected corpus was evaluated using corpus homogeneity and similarity scores.

In order to ensure that the Web corpus is sufficiently on-topic, it is important to know how similar the two corpora are. Corpus similarity also plays a pivotal role when porting an NLP application from one domain with one corpus to another domain with a different corpus. Corpus similarity is a complex issue and there is no generally accepted method to measure corpus similarity; (Kilgarriff, 1997; Kilgarriff and Rose, 1998 and Kilgarriff, 2001) argued that it is most important to first determine the homogeneity of a corpus before computing its similarity to another corpus, as the

[30] http://www.nlm.nih.gov/
[31] http://bootcat.sslmit.unibo.it/

judgement of similarity can become unreliable if a homogenous corpus is compared with a heterogeneous one. Kilgarriff (1997) presented an overview of various approaches for corpus similarity and proposed a word frequency list approach to measure corpus similarity and homogeneity. We used the Kilgarriff (1997) approach as it is considerably easier to count words accurately rather than syntactic categories.

In order to measure corpus homogeneity, we divided the corpus into two equal parts and produced a word frequency list of each sub-corpus by processing the text using GENIA tagger and filtering out punctuations and stop words. In the next step we took the 500 most frequent words from each sub-corpus and calculated the chi-square statistics for the difference between two sub-corpora, as Kilgarriff and Rose (1998) and Kilgarriff (2001) showed that chi-square statistics perform considerably better than other information-theoretic and statistical measures. To determine the similarity between the two corpora, we also produced the top 500 words from each corpus and calculated the chi-square statistics for each corpus. Low chi-square scores indicate homogeneous and highly similar corpora, while high scores correspond to heterogeneous corpora and dissimilar corpora.

Corpus	Chi-Score
GENIA	1379.693
GENIA EVENT	2364.577
WEB	14750.369
BNC	20872371.995

Table 9: Homogeneity scores of corpora

Table 9 shows the homogeneity scores between two sub-corpora in each corpus we used in the experiment. We observe that GENIA and GENIA EVENT corpora achieve quite a low score which in turn shows that both these two corpora are homogenous. This is rather unsurprising as both corpora were compiled by hand to ensure topic relevance and are generally accepted as benchmark biomedical corpora. WEB and BNC scores show that these two corpora are more heterogeneous. BNC exhibits the greatest heterogeneity, which is obviously explained by the fact that the corpus is meant to cover the broadest possible range of domains in general British

English. The WEB corpus is much more homogeneous than BNC, but still has a chi-square score of magnitude greater than the GENIA corpora, reflecting the fact that automatic web collection methods are still incapable of ensuring the same level of topic relevance as achieved in manually compiled corpora.

In the next step, we will calculate the similarity scores between these corpora using Chi-Score. Table 10 shows similarity scores in which GENIA and GENIA EVENT corpora are quite similar to each other while in the case of all other corpora the high score means that they are quite dissimilar to each other.

	GENIA EVENT	WEB	BNC
GENIA	2137.63	173207.002	23686564.063
GENIA EVENT		136568.630	23008298.781
WEB			28068572.14

Table 10: Similarity scores of corpora

As mentioned earlier that BNC is a heterogeneous corpus, which is also reflected here too in the form of a higher similarity score while the WEB corpus similarity score is also quite high due to a higher homogenous score when compared to the manually compiled corpora of GENIA and GENIA EVENT respectively.

We collected the Web corpus to attain higher recall in our experiments but as is quite obvious from the homogeneity and similarity scores (Table 9 and 10), the Web corpus is not homogenous and also not similar to GENIA or GENIA EVENT corpus. One of the possible reasons for this is that GENIA is a very narrow-domain corpus and it is hard to collect relevant topical documents automatically.

3.1.5.2 Evaluation method

In order to evaluate the quality of the extracted patterns, we examined their ability to capture pairs of related named entities in the manually annotated evaluation corpus, without recognising the types of the semantic relations. Selecting a certain number of best-ranking patterns, we measured precision, recall and F-score.

To test the statistical significance of differences in the results of different methods and configurations, we used a paired t-test, having randomly divided the evaluation corpus (GENIA EVENT Annotation corpus) into 20 subsets of equal size; each subset containing 461 sentences on average. We collected precision, recall and F-score for each of these subsets and then using paired t-test we found statistical significance between different surface pattern types and also between different ranking methods using score-thresholding measure.

3.1.6 Results

In the first phase of experiments, we considered all surface pattern types (e.g. untagged, PoS and verb-centred) with out prepositions. We carried out our experiments on all 3 corpora (GENIA, WEB and GENIA+ WEB) for all three surface pattern types. As we found in Section 3.1.5.1 that the WEB corpus is not similar to the GENIA or the GENIA EVENT corpus, in this section we will discuss the results for the GENIA corpus only while Appendix C contains complete results for all three corpora along with precision, recall and F-scores.

The numbers of untagged word patterns extracted from each corpus are: GENIA 12230, WEB 42718, GENIA+WEB 52511, BNC 1956473 and GENIA EVENT 5763. Figure 4 shows the rank-thresholding results for untagged word patterns using the GENIA corpus. Precision scores are represented along the Y-axis; recall is very low in rank-thresholding measure (see Table 1 in Appendix C for complete results in terms of precision, recall and F-scores).

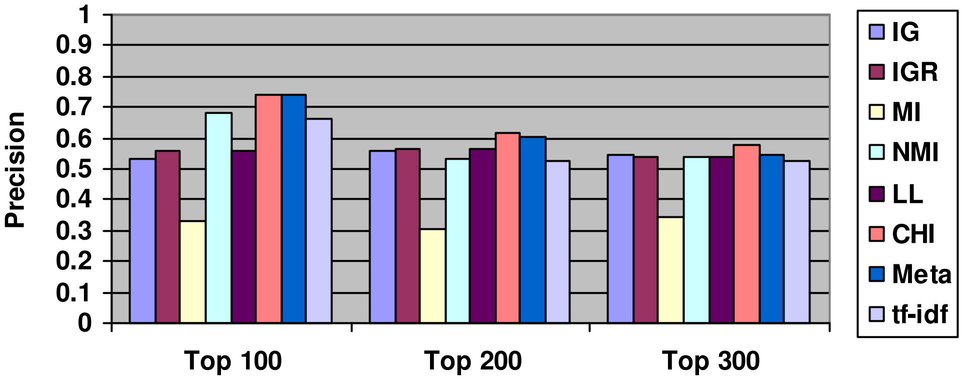


Figure 4: Rank-thresholding results for untagged word patterns using GENIA corpus

Figure 4 clearly shows that CHI, Meta and NMI are the best performing ranking methods while MI is the worst. Moreover, IG, IGR and LL achieved quite similar results.

After rank-thresholding, the next set of experiments is based on the score-thresholding measure for untagged word patterns for each corpus (e.g. GENIA, WEB and GENIA+WEB). Here we are considering only those threshold scores which enable us to attain high precision scores (see Table 4 in Appendix C for complete results in terms of precision, recall and F-score for each corpus). Figure 5 shows the results of score-thresholding measures for untagged word patterns using GENIA corpus.

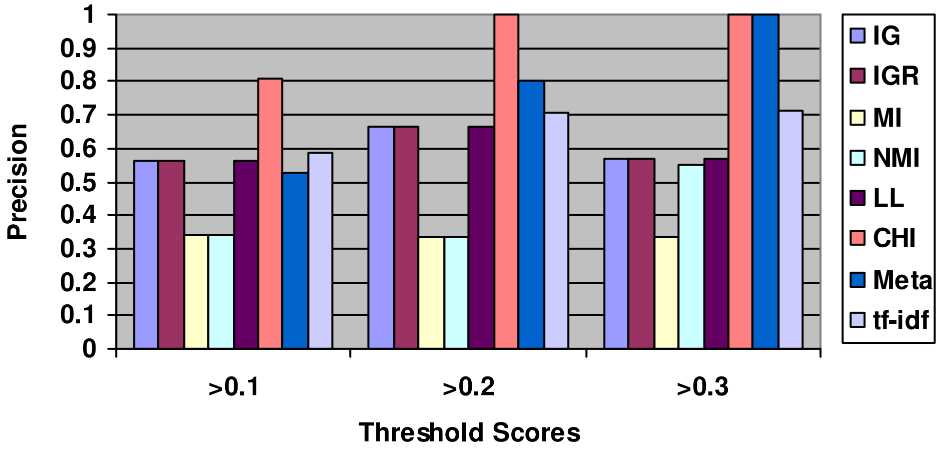


Figure 5: Score-thresholding results for untagged word patterns using GENIA corpus

In Figure 5, we are able to achieve 100% precision scores using CHI and Meta ranking methods but at the cost of a very low recall. Here too, IG, IGR and LL achieved quite similar results while tf-idf performed better than them.

We carried out a similar set of experiments using PoS-tagged word patterns. The numbers of PoS-tagged word patterns extracted from each corpus are: GENIA 12239, WEB 43708, GENIA+WEB 53871, BNC 1969040 and GENIA EVENT 5676. Figure 6 shows the rank-thresholding results for PoS-tagged word patterns using the GENIA corpus with precision scores are represented along the Y-axis (see Table 2 in Appendix C for complete results in terms of precision, recall and F-score for each corpus).

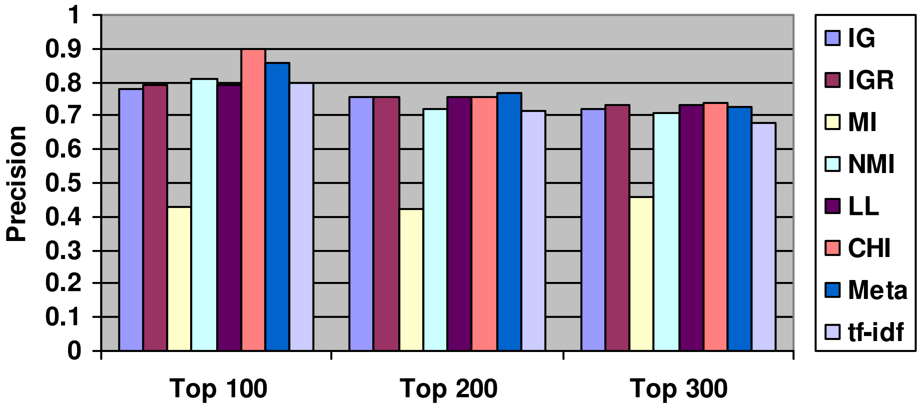


Figure 6: Rank-thresholding results for PoS-tagged word patterns using GENIA corpus

The results in Figure 6 indicate that similar to Figure 4 (rank-thresholding results of untagged word patterns) CHI, Meta and NMI are the best performing ranking methods while MI is the worst. The overall results obtained using the rank-thresholding measure in PoS-tagged word patterns show that it is able to achieve higher precision scores than compared to untagged word patterns (Figure 4).

The next set of experiments is based on the score-thresholding measure for PoS-tagged word patterns for each corpus. Similar to untagged word patterns we are only reporting those threshold scores for the GENIA corpus that attain high precision scores (see Table 5 in Appendix C for complete results in terms of precision, recall and F-score for each corpus).

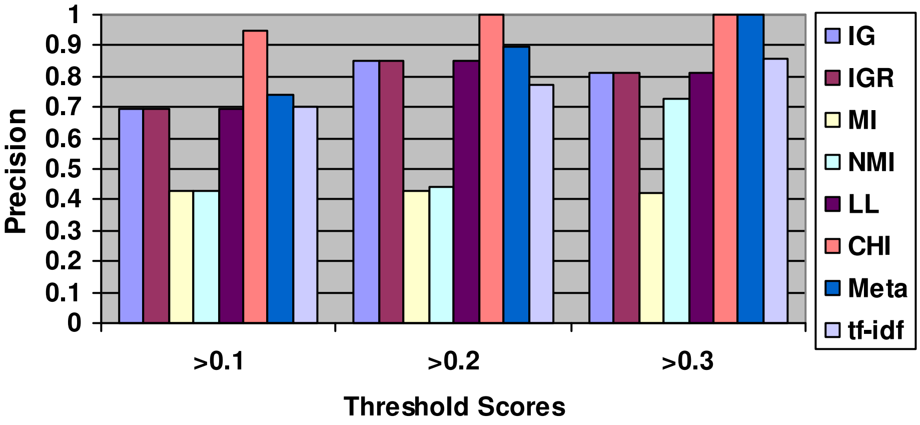


Figure 7: Score-thresholding results for PoS-tagged word patterns using GENIA corpus

Similar to Figure 5 here in Figure 7 too we are able to achieve 100% precision score but recall is very low.

In the final set of experiments of surface type patterns without prepositions, we carried out a similar set of experiments using verb-centred word patterns. The numbers of verb-centred word patterns extracted from each corpus are: GENIA 8328, WEB 28645, BNC 1604809 and GENIA EVENT 4010. Figure 8 shows the rank-thresholding results for verb-centred word patterns using the GENIA corpus, precision scores are represented along the Y-axis (see Table 3 in Appendix C for complete results in terms of precision, recall and F-score for each corpus).

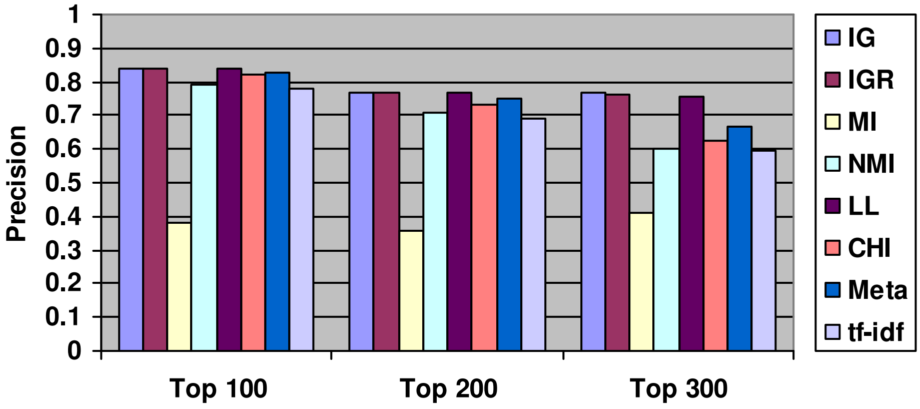


Figure 8: Rank-thresholding results for verb-centred word patterns using GENIA corpus

The overall results achieved using the rank-thresholding measure in verb-centred word patterns indicate that it is similar to PoS-tagged word patterns in the way that it is able to achieve higher precision scores than compared to untagged word patterns (Figure 4). Moreover, similar to other surface pattern types here too IG, IGR and LL attained quite similar results in all three corpora.

In the next set of experiments, we used score-thresholding measure for verb-centred word patterns for each corpus using only those threshold scores that provide us higher precision scores for the GENIA corpus (see Table 6 in Appendix C for complete results in terms of precision, recall and F-score for each corpus).

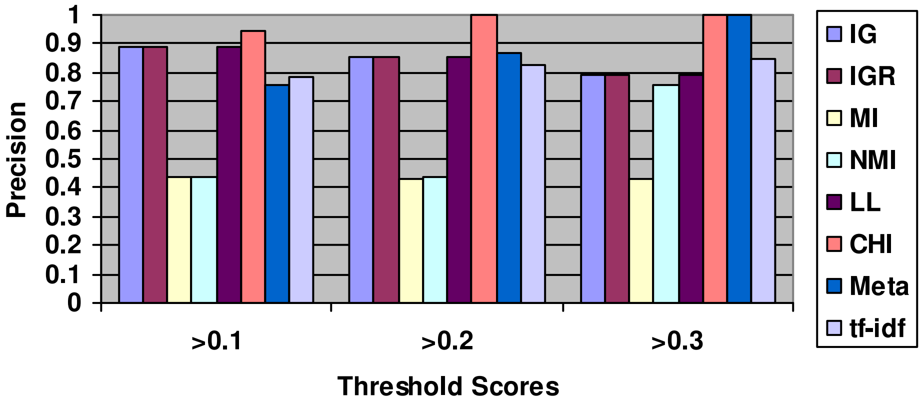


Figure 9: Score-thresholding results for verb-centred word patterns using GENIA corpus

Figure 9 shows the results of the score-thresholding measure in verb-centred word patterns using the GENIA corpus and they indicate that overall we are able to achieve higher precision scores than compared to other surface pattern types for the GENIA corpus.

In the next phase of experiments, we also considered prepositions present between two NEs along with the content words during the pattern learning process and again obtain the same surface pattern types (i.e. untagged word patterns, PoS-tagged word patterns and verb-centred word patterns) along with prepositions. Prepositions are used to express relations of place, direction, time or possessions. We used the same set of corpora and ranking methods as used in previous phase of experiments.

Similar to the first phase of experiments, we carried out our experiments on all three corpora for each surface pattern type with prepositions. The numbers of untagged word patterns along with prepositions extracted from each corpus are: GENIA 10093, WEB 34122, GENIA+WEB 41990, BNC 991004 and GENIA EVENT 4854. Figure 10 shows the rank-thresholding results for untagged word patterns along with prepositions using the GENIA corpus (see Table 7 in Appendix C for complete results in terms of precision, recall and F-score for each corpus).

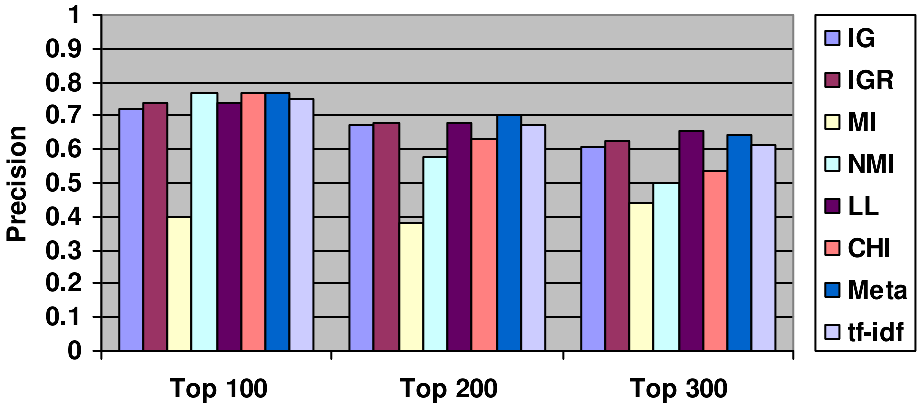


Figure 10: Rank-thresholding results for untagged word patterns along with prepositions using GENIA corpus

The results in Figure 10 show that addition of prepositions in untagged word patterns has been very useful and has increased overall precision scores compared with untagged word patterns without prepositions for GENIA corpus (Figure 4).

After rank-thresholding, the next set of experiments is based on the score-thresholding measure for untagged word patterns along with prepositions for each corpus (e.g. GENIA, WEB and GENIA+WEB). Here too we are considering only those threshold scores which enable us to attain high precision scores for the GENIA corpus (see Table 10 in Appendix C for complete results in terms of precision, recall and F-score for each corpus).

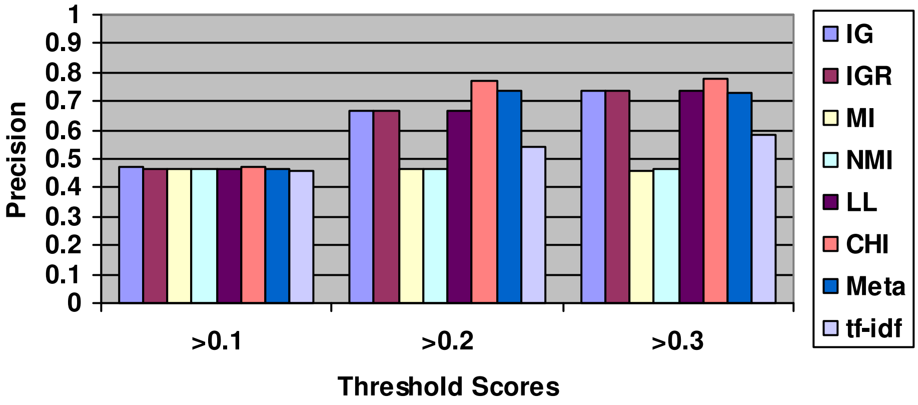


Figure 11: Score-thresholding results for untagged word patterns along with prepositions using GENIA corpus

We carried out a similar set of experiments using PoS-tagged word patterns along with prepositions. The numbers of PoS-tagged word patterns along with prepositions extracted from each corpus are: GENIA 9237, WEB 33871, GENIA+WEB 41245, BNC 840057 and GENIA EVENT 4446. Figure 12 shows the rank-thresholding results for PoS-tagged word patterns along with prepositions using the GENIA corpus, precision scores are along the Y-axis (see Table 8 in Appendix C for complete results in terms of precision, recall and F-score for each corpus).

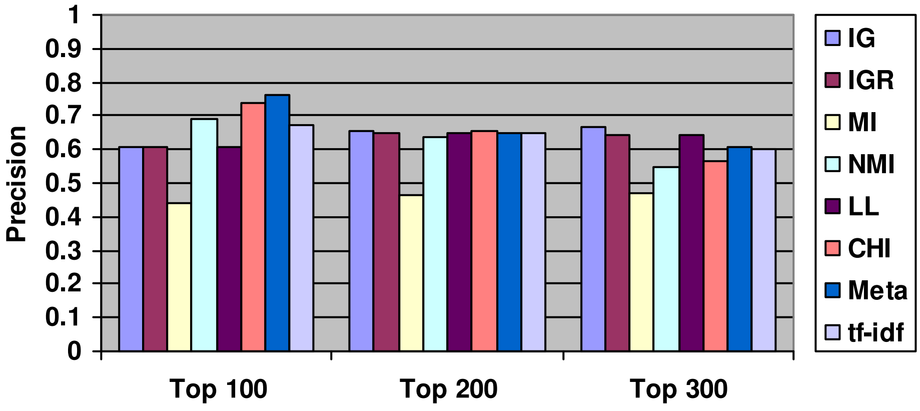


Figure 12: Rank-thresholding results for PoS-tagged word patterns along with prepositions using GENIA corpus

After rank-thresholding, in the next set of experiments we used score-thresholding measure for PoS-tagged word patterns along with prepositions for each corpus (see Table 11 in Appendix C for complete results in terms of precision, recall and F-score for each corpus). Figure 13 shows the results of the score-thresholding measure for

PoS-tagged word patterns along with prepositions using the GENIA corpus and they indicate that additions of prepositions in PoS-tagged word patterns has been very helpful and has increased overall precision scores compared with PoS-tagged word patterns without prepositions (Figure 7).

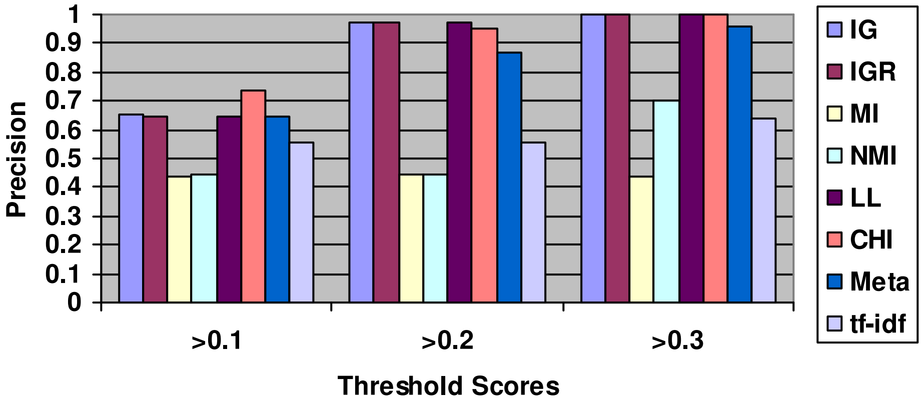


Figure 13: Score-thresholding results for PoS-tagged word patterns along with prepositions using GENIA corpus

We carried out a similar set of experiments using verb-centred word patterns along with prepositions for each corpus. The numbers of verb-centred word patterns along with prepositions extracted from each corpus are: GENIA 6645, WEB 23931, GENIA+WEB 29353, BNC 598948 and GENIA EVENT 3271. Figures 14 shows the rank-thresholding results for verb-centred patterns along with prepositions using GENIA, precision scores are along the Y-axis (see Table 9 in Appendix C for complete results in terms of precision, recall and F-scores for each corpus).

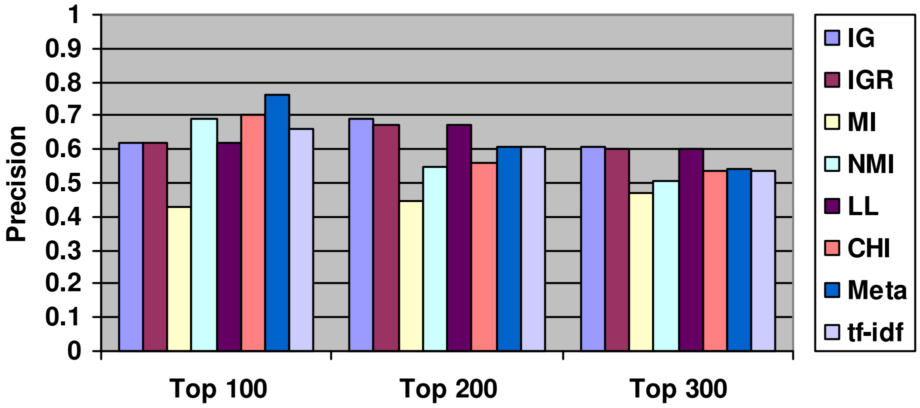


Figure 14: Rank-thresholding results for verb-centred word patterns along with prepositions using GENIA corpus

After rank-thresholding, we used score-thresholding measure for verb-centred surface patterns along with prepositions for each corpus (see Table 12 in Appendix C for complete results in terms of precision, recall and F-score for each corpus). Figure 15 shows the results of the score-thresholding measure for verb-centred patterns with prepositions for the GENIA corpus, precision scores are represented along the Y-axis.

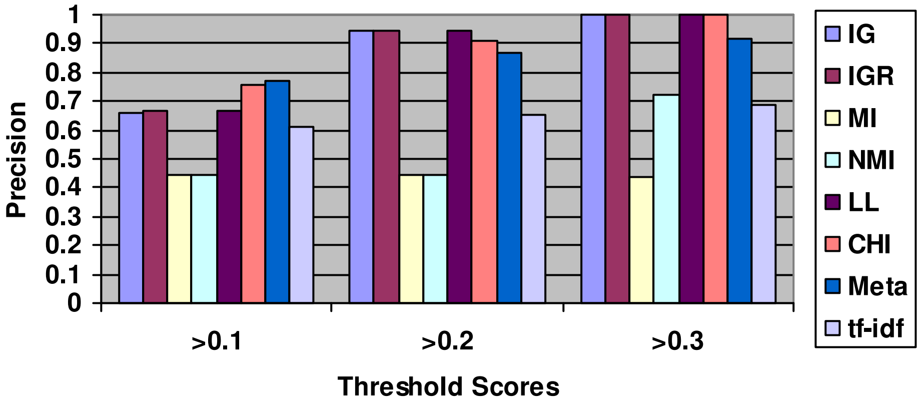


Figure 15: Score-thresholding results for verb-centred word patterns along with prepositions using GENIA corpus

3.1.6.1 Ranking methods

In Section 3.1.6, we carried out our experiments on three different surface pattern types (untagged, PoS-tagged and verb-centred) without prepositions and with prepositions. We used different pattern ranking methods (see Section 3.1.4) and in all experiments we found that IG, IGR and LL achieved quite similar results while CHI, Meta and NMI are the best performing ranking methods while MI is the worst in terms of precision scores. The tf-idf ranking method performed better than MI on all occasions but it is not really applicable to our work as our corpus consists of those documents that describe relevant domain information only as compared to the corpus used by Sudo et al. (2003). Even though CHI and Meta ranking methods attained higher precision scores but recall scores are very low. We used two evaluation measures: rank-thresholding and score-thresholding, we found that score-thresholding is a better performing measure than rank-thresholding as we are able to achieve 100% precision score with it. Moreover, when we compared different surface pattern types without prepositions to different surface pattern types with prepositions, we found that generally surface pattern types with preposition performed better as the addition of prepositions is useful for extracted semantic relations. We explored three surface pattern types (untagged, PoS-tagged and verb-centred) and found that verb-centred and PoS-tagged pattern types are better than untagged word patterns. Figure 16 shows the precision score of the best performing ranking method (CHI-score) for each corpus in verb-centred patterns in the score-thresholding measure while Figure 17 shows the same results for verb-centred patterns along with prepositions.

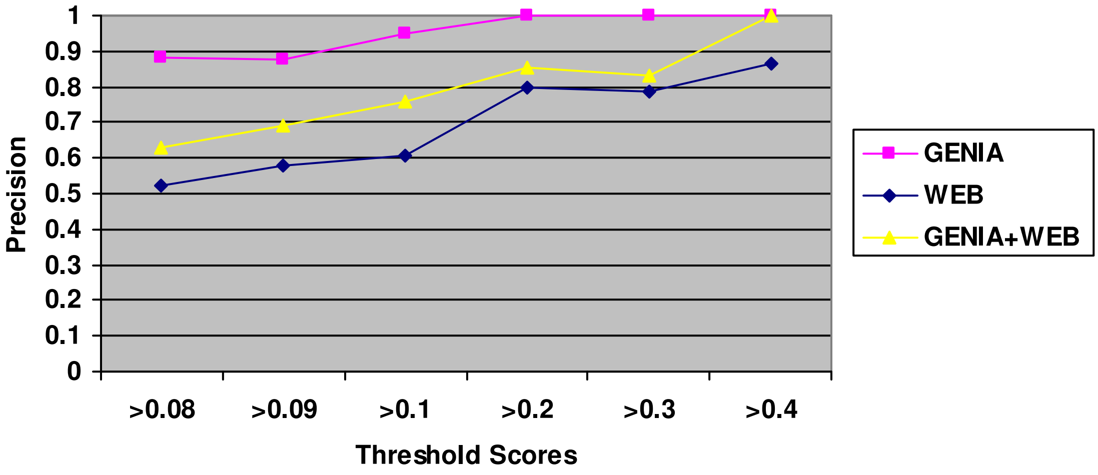


Figure 16: Precision scores of best performing ranking method for verb-centred patterns in score-thresholding

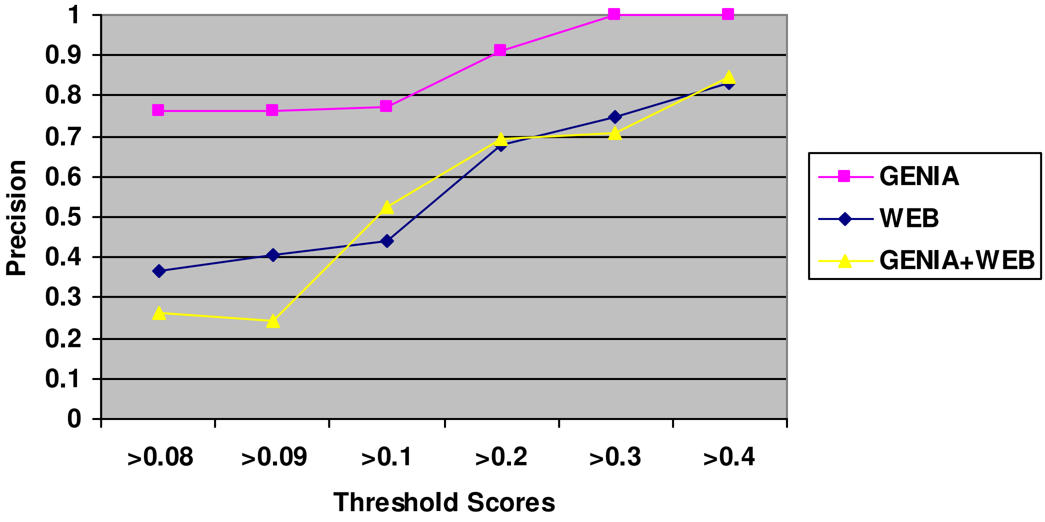


Figure 17: Precision scores of best performing ranking method for verb-centred patterns with prepositions in score-thresholding

Overall in all these sets of experiments, IG, IGR and LL ranking methods perform quite similarly to each other and in general, there is no statistical significant difference between them. While literature on the topic suggests that IGR performs better than IG (Quinlan, 1986; Manning and Schütze, 1999), we found that in general there is no statistical significant difference between IG and IGR, IGR and LL in all three patterns types. Moreover, in all these experiments, obviously due to the aforementioned problem, MI performs quite poorly; the normalised version of MI helps to alleviate this problem. Moreover, there exists a statistically significant difference ($p < 0.01$) between NMI and the other ranking methods in all three pattern types. The meta-ranking method did not improve on the best individual ranking method as expected. Moreover, we found that there is a statistically significant difference ($p < 0.05$) between the meta-ranking method and all the other ranking methods for all three pattern types. We also found that the score-thresholding method is better than the rank-thresholding method as we were able to achieve 100% precision scores.

3.1.6.2 Types of patterns

PoS-tagged word patterns and verb-centred patterns perform better than untagged word patterns. Verb-centred patterns work well, because verbs are known to express semantic relations between named entities using syntactic arguments to the verb; PoS-tagged word patterns add important semantic information into the pattern and possibly disambiguate words appearing in the pattern.

In order to find out that whether the differences between the three patterns types are statistically significant, we carried out a paired t-test again. We found that there is no statistically significant difference between PoS-tagged word patterns and verb-centred patterns. Apart from IG, IGR and LL there is a statistically significant difference between all the ranking methods of untagged word patterns and PoS-tagged word patterns, untagged word patterns and verb-centred patterns respectively.

3.1.6.3 Precision vs. F-measure optimisation

In terms of F-score verb-centred patterns achieved a higher F-score as compared to other pattern types while the addition of prepositions in each pattern type also results in a higher F-score (see Appendix C for more details). Moreover CHI and NMI are the best performing ranking methods, Figure 18 and Figure 19 show precision, recall and F-score for verb-centred patterns with prepositions for the GENIA corpus achieved using these ranking methods.

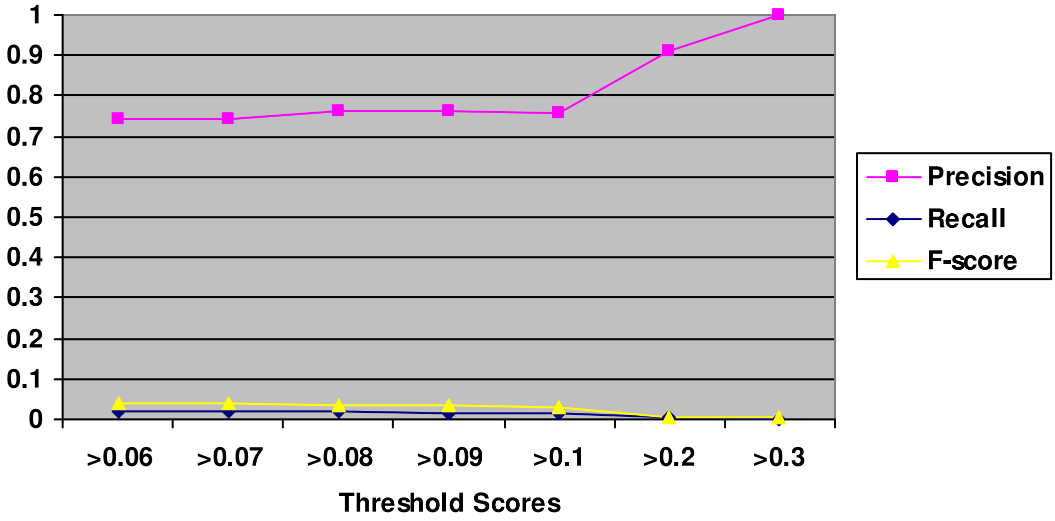


Figure 18: Precision, recall and F-score for verb-centred patterns with prepositions in score-thresholding measure using CHI

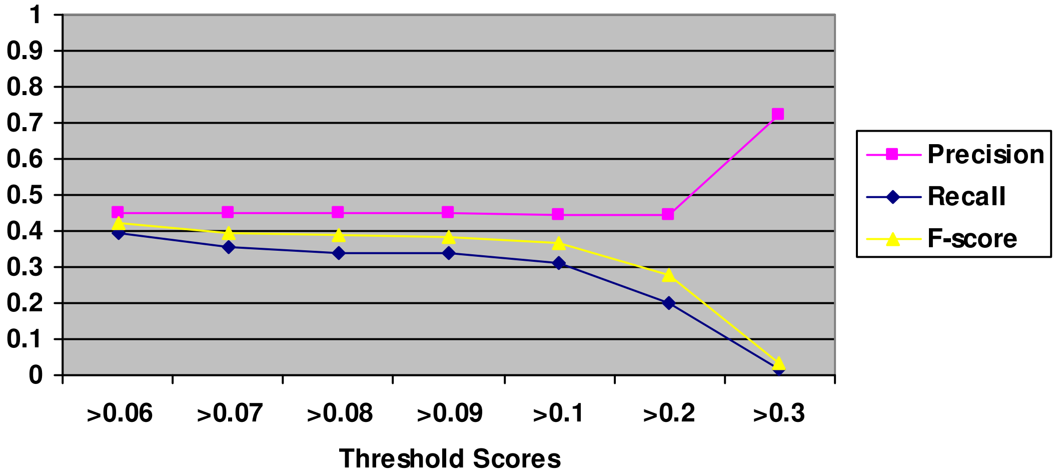


Figure 19: Precision, recall and F-score for verb-centred patterns with prepositions in score-thresholding measure using NMI

Figure 18 clearly shows that even though CHI achieved high precision scores, recall and F-score are quite low while Figure 19 shows that NMI achieved much better recall and F-score than CHI but at the cost of low precision scores.

The score-thresholding measure achieves higher precision than the rank-thresholding measure. High precision is quite important in applications such as MCQ generation. In score-thresholding, it is possible to optimise for high precision (up to 100%), though the F-score is generally quite low. MCQ applications rely on the production of good questions rather than the production of all possible questions, so high precision plays a vital role in such applications.

3.2 Unsupervised Dependency-based Patterns

3.2.1 Automatic Parsing of Text

Syntactic analysis of text, also known as parsing, is a process of determining the grammatical structure of its sentence constituents. In syntactic analysis, a sentence is recursively decomposed into smaller units called constituents or phrases. These constituents are then categorised into noun phrases or verb phrases according to their internal structures. Syntactic analysis is generally represented in the form of a parse tree. Syntax plays an important role in making language useful for communication. Syntax in linguistics attempts to describe the language in terms of certain rules. In relation to automatic parsing, many theoretical approaches are presented so far in the area of syntax.

Dependency trees are regarded as a suitable basis for semantic pattern acquisition as they abstract away from the surface structure to represent relations between elements (entities) of a sentence. Semantic patterns represent semantic relations between elements of sentences. One of the advantages of using dependency trees is that they provide a useful structure for the sentences by annotating edges with dependency functions e.g. subject, object etc. (Fundel et al., 2007). In a dependency tree a pattern is defined as a path in the dependency tree passing through zero or more intermediate nodes within a dependency tree (Sudo et al., 2001). Stevenson and Greenwood (2009) provided an insight of the usefulness of dependency patterns in their work (see Section 2.6.3.6). In their work, they revealed that dependency parsers have the advantage of generating analyses which abstract away from the surface realisation of text to a greater extent than phrase structure grammars tend to, resulting in semantic information being more accessible in the representation of the text which can be useful for IE.

Several approaches in IE have relied on dependency trees in order to extract patterns for the automatic acquisition of IE systems (Yangarber et al., 2000; Sudo et al., 2001; Sudo et al., 2003; Stevenson and Greenwood, 2005 and Greenwood et al., 2005) (see Sections 2.6.3). Apart from IE, Lin and Pantel (2001) used dependency trees in order to infer rules for question answering while Szpektor et al. (2004) had made use of dependency trees for paraphrase identification. Moreover, dependency parsers are used most recently in the systems which identify protein interactions in biomedical texts (Katrenko and Adriaans, 2006; Erkan et al., 2007 and Saetre et al., 2007).

All the abovementioned approaches have used different pattern models based on the particular part of the dependency analysis. The motive behind all of these models is to extract the necessary information from text without being overly complex. All of the pattern models have made use of the semantic patterns based on the dependency trees for the identification of items of interest in text. These models vary in terms of their complexity, expressivity and performance in an extraction scenario.

3.2.2 Our Approach

In our dependency-based approach, we employed two dependency tree pattern models: SVO pattern model (SVO patterns) and an adapted version of the linked chain pattern model. We used SVO pattern model (Yangarber et al., 2000; see Section 2.6.3.1 for more details) as a baseline in our experiments. In the SVO model, we extracted all subject-verb-object tuples from the dependency parse of a sentence and discarded the remainder of the dependency parse.

Our adapted linked chain pattern model approach (Afzal et al., 2011) is based on the linked chain pattern model presented by Greenwood et al. (2005) (see Section 2.6.3.4). Linked chain pattern model combines the pairs of chain in a dependency tree which share common verb root but no direct descendants. We selected the linked chain dependency pattern model as it is the best performing pattern model and its performance is consistently better than the collective performance of both SVO and chain dependency pattern models (Stevenson and Greenwood, 2009).

In our approach, we have treated every Named Entity (NE) as a chain in a dependency tree if it is less than 5 dependencies away from the verb root and the word linking the NEs to the verb root are from the category of content words (Verb, Noun, Adverb and Adjective) along with prepositions. We consider only those chains in the dependency tree of a sentence which contain NEs, which is much more efficient than the subtree model of Sudo et al. (2003) (see Section 2.6.3.3), where all subtrees containing verbs are taken into account. This allows us to extract more meaningful patterns from the dependency tree of a sentence. We extract all NE chains which follow aforementioned rule from a sentence and combine them together. The extracted patterns are then stored in a database along with their frequencies.

3.2.3 Extraction of Candidate Patterns

As with the learning of surface-based patterns, our general approach to learn dependency-based patterns consists of the same two main stages: (i) the construction of potential patterns from an unannotated domain corpus and (ii) their relevance ranking.

3.2.3.1 Pre-processing steps

The first step in constructing candidate patterns is to perform NE recognition in an unannotated domain corpus. We will explain the whole process of candidate patterns extraction from the dependency trees with the help of an example shown below:

Fibrinogen activates NF-kappa B in mononuclear phagocytes.

We used the GENIA[32] tagger for NER and the following example shows the NER from a biomedical text:

<protein> Fibrinogen </protein> activates <protein> NF-kappa B </protein> in <cell_type> mononuclear phagocytes </cell_type>.

Once the NEs are recognised in the domain corpus by the GENIA tagger, we replace all the NEs with their semantic class respectively, so the aforementioned sentence is transformed into the following sentence.

PROTEIN activates PROTEIN in CELL.

The transformed sentences are then parsed by using the Machinese Syntax[33] parser (Tapanainen and Järvinen, 1997). The Machinese Syntax parser uses a functional dependency grammar for parsing. The Machinese Syntax parser first labels each word with its all possible function types and then applies a collection of handwritten rules to introduce links between specific types in a given context and remove all the other function types. The Machinese Syntax parser was evaluated in terms of correct identification of attached heads on three different genres in the Bank of English (Järvinen, 1994) data. Table 11 shows the results in terms of precision and recall.

	Precision	**Recall**
Broadcast	93.4%	88.0%
Literature	96.0%	88.6%
Newspaper	95.3%	87.9%

Table 11: Percentages of heads correctly attached

Stevenson and Greenwood (2007) used three different parsers including the Machinese Syntax parser in order to compare different IE models. They carried out their experiments on two different corpora: MUC-6 corpus and a biomedical corpus (see Section 2.6.3.6).

[32] http://www-tsujii.is.s.u tokyo.ac.jp/GENIA/tagger/
[33] http://www.connexor.com/software/syntax/

Figure 20 shows the dependency tree produced by the parser for the aforementioned adapted sentence example.

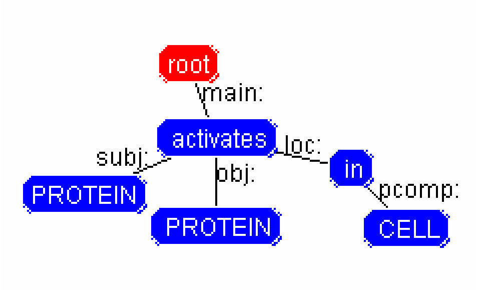


Figure 20: Dependency tree of *'PROTEIN activates PROTEIN in CELL'*

The analyses produced by the Machinese Syntax parser are encoded to make the most of information they contain and ensure consistent structures from which patterns could be extracted. Figure 21 shows the encoded output of a biomedical text.

```
<s id="S1">
<W ID="2" LEMMA="protein" POS="N" FUNC="SUBJ" DEP="3">PROTEIN</W>
<W ID="3"LEMMA="activate" POS="V" FUNC="+FMAINV"DEP="1">activates</W>
<W ID="4" LEMMA="protein" POS="N" FUNC="OBJ" DEP="3">PROTEIN</W>
<W ID="5" LEMMA="in" POS="PREP" FUNC="ADVL" DEP="3">in</W>
<W ID="6" LEMMA="cell" POS="N" FUNC="P" DEP="5">CELL</W>
<W ID="7" LEMMA="." POS="" FUNC="" DEP="none">.</W>
</s>
```

Figure 21: Encoded biomedical text

3.2.3.2 Dependency-based patterns

After the encoding, the patterns are extracted from the dependency trees using the methodology described in Section 3.2.2. For example the following SVO pattern was extracted from the Figure 21.

[V/activate] (subj[PROTEIN] + obj[PROTEIN])

The following adapted linked chain patterns were extracted from the same example (Figure 21):

[V/activate] (subj[PROTEIN] + obj[PROTEIN])

[V/activate] (obj[PROTEIN] + prep[in] + p[CELL_TYPE])

For dependency tree patterns representation, we employed a similar sort of formalism to that used by Sudo et al. (2003). Each node in the dependency tree is represented in the format *a[B]* e.g. *subj[PROTEIN]* where *a* is the dependency relation between this node and its parent *(subj)* and *B* is the semantic class of the named entity. The relationship between nodes is represented as *X (A+B+C)* which indicates that nodes *A, B* and *C* are direct descendants of *X*. The patterns along with their frequencies are stored in a database. Similar to surface-based patterns, we also filtered out the patterns containing only stop-words in dependency-based patterns too. In SVO patterns, we extracted only those SVO patterns where both subject and object are named entities. Table 12 shows some examples of the most frequent SVO patterns along with their frequencies extracted from the GENIA corpus.

Patterns	Frequency
[V/contain] (subj[DNA] + obj[DNA])	34
[V/activate] (subj[PROTEIN] + obj[PROTEIN])	32
[V/contain] (subj[PROTEIN] + obj[PROTEIN])	19
[V/induce] (subj[PROTEIN] + obj[PROTEIN])	18
[V/encode] (subj[DNA] + obj[PROTEIN])	17
[V/express] (subj[CELL_TYPE] + obj[PROTEIN])	16
[V/inhibit] (subj[PROTEIN] + obj[PROTEIN])	14
[V/form] (subj[PROTEIN] + obj[PROTEIN])	6
[V/regulate] (subj[PROTEIN] + obj[PROTEIN])	6
[V/stimulate] (subj[PROTEIN] + obj[PROTEIN])	6

Table 12: SVO patterns along with their frequencies

The total numbers of SVO patterns extracted from the GENIA corpus is very small and one of the main reasons for this is that the SVO pattern model does not perform well in the biomedical domain. This fact was also highlighted by Stevenson and Greenwood (2009) and they argued that the reason behind this is that in the biomedical domain named entities are described in ways that the SVO pattern model is unable to represent as it is restricted to verbs and their direct arguments only. In their work they compared various pattern models using two domains: MUC-6 and biomedical data (see Section 2.6.3.6)

Table 13 shows some examples of the most frequent adapted linked chain patterns along with their frequencies extracted from the GENIA corpus.

Patterns	Frequency
[V/contain] (subj[DNA] + obj[DNA])	34
[V/activate] (subj[PROTEIN] + obj[PROTEIN])	32
[V/contain] (subj[PROTEIN] + obj[PROTEIN])	19
[V/induce] (subj[PROTEIN] + app[PROTEIN])	19
[V/activate] (a[DNA] + obj[PROTEIN])	18
[V/induce] (subj[PROTEIN] + obj[PROTEIN])	18
[V/interact] (subj[PROTEIN] + prep[in] + p[PROTEIN])	17
[V/induce] (subj[PROTEIN] + obj[phosphorylation] + prep[of] + p[PROTEIN])	17
[V/encode] (subj[DNA] + obj[PROTEIN])	17
[V/induce] (subj[PROTEIN] + subj[PROTEIN])	17

Table 13: Adapted linked-chain patterns along with their frequencies

In our experiments we preferred to use an adapted linked chain pattern model as it is possible to encode more of the information present in a sentence than compared to SVO pattern model (Section 2.6.3.1) or chain pattern model (Section 2.6.3.2) and this fact was also highlighted by Stevenson and Greenwood (2009). Moreover, SVO pattern model performed very poorly in the biomedical domain as compared to linked chain pattern model (Stevenson and Greenwood, 2009).

3.2.4 Pattern Ranking

In order to rank extracted candidate patterns, we employed the same information theoretic concepts: Information Gain (IG), Information Gain Ratio (IGR), Mutual Information (MI), Normalised Mutual Information (NMI) and statistical tests of association: Log-likelihood (LL) and Chi-Square (CHI), along with meta-ranking and tf-idf ranking methods which we used in the surface-based approach (see Section 3.1.4 for further details).

3.2.5 Evaluation

We used the same experimental data as used in the surface-based patterns experiments (see Section 3.1.5 for further details). The numbers of adapted linked chain dependency patterns extracted from each corpus are: GENIA 5066, WEB 13653, GENIA+WEB 17694, BNC 419274 and GENIA EVENT 3031. The quality of extracted patterns is evaluated by employing the same approach as described in Section 3.1.5.2.

3.2.6 Results

We conducted our experiments on all 3 corpora (GENIA, WEB and GENIA+WEB). Similar to the surface-based approach, here we will discuss the results for the GENIA corpus only while the complete results for all three corpora in terms of precision, recall and F-scores are given in Appendix C. Figure 22 shows the rank-thresholding results for adapted linked chain dependency patterns using the GENIA corpus. Here precision scores are represented along the Y-axis (for complete results see Table 13 in Appendix C).

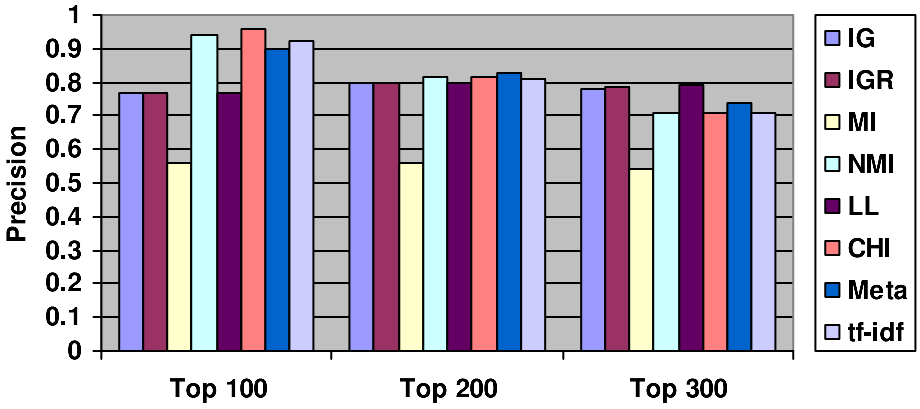


Figure 22: Rank-thresholding results for adapted linked chain patterns using GENIA corpus

Figure 22 shows that similar to the surface-based approach CHI and NMI are the best performing ranking methods while MI is the worst. Moreover, IG, IGR and LL achieved quite similar results.

In the next step we used score-thresholding measure for each corpus similar to the surface-based approach. Here too, we are considering only those threshold scores that give us high precision scores (see Table 14 in Appendix C for complete results for each corpus). Figure 23 shows the results of score-thresholding measures for adapted linked chain dependency patterns using the GENIA corpus.

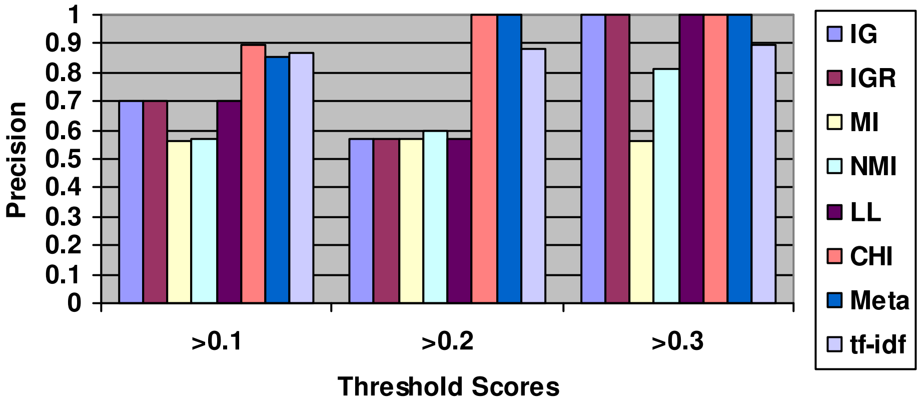


Figure 23: Rank-thresholding results for adapted linked chain patterns using GENIA corpus

3.2.6.1 Ranking methods

We carried out our experiments using both ranking measures: rank-thresholding and score-thresholding. In both set of experiments, similar to the surface-based approach (Section 3.1.6) CHI is the best performing ranking method but recall scores are very low. MI is the worst performing ranking method while IG, IGR and LL attained quite similar results. Moreover, we found that there is a no statistical significant difference ($p < 0.05$) between IG and LL, IGR and LL. Similar to the surface-based approach tf-idf achieved quite reasonable results but it is not the best performing ranking method. Figure 24 shows the precision scores of the best performing ranking method (CHI) in the score-thresholding method for dependency patterns.

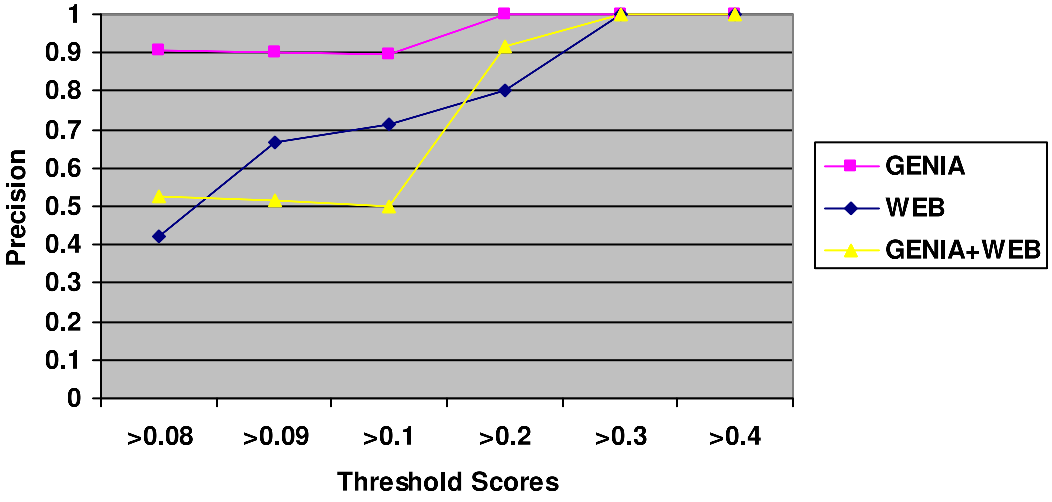


Figure 24: Precision scores of best performing ranking method for adapted linked chain dependency patterns in score-thresholding

3.2.6.2 Score vs. rank thresholding

We also found that the score-thresholding method produces better results than the rank-thresholding as we are able to achieve higher precision with the former measure.

3.2.6.3 Precision vs. F-measure optimisation

As mentioned earlier, CHI is the best performing ranking method in terms of precision scores while recall scores are very low. Using NMI ranking method we are able to achieve quite reasonable results in terms of both precision and recall scores. Figure 25 and Figure 26 show precision, recall and F-score for the GENIA corpus using these ranking methods (CHI and NMI).

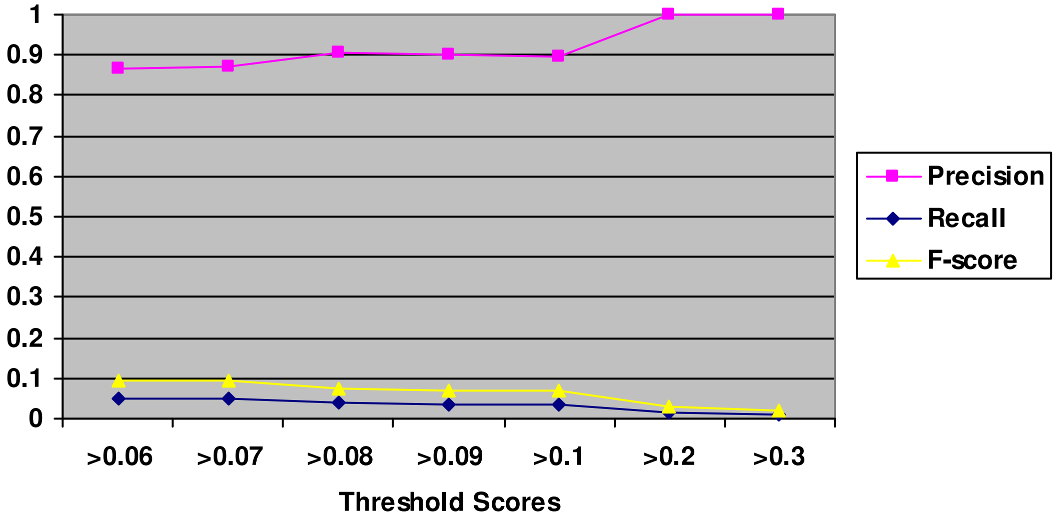


Figure 25: Precision, recall and F-score for adapted linked chain dependency patterns in score-thresholding measure using CHI

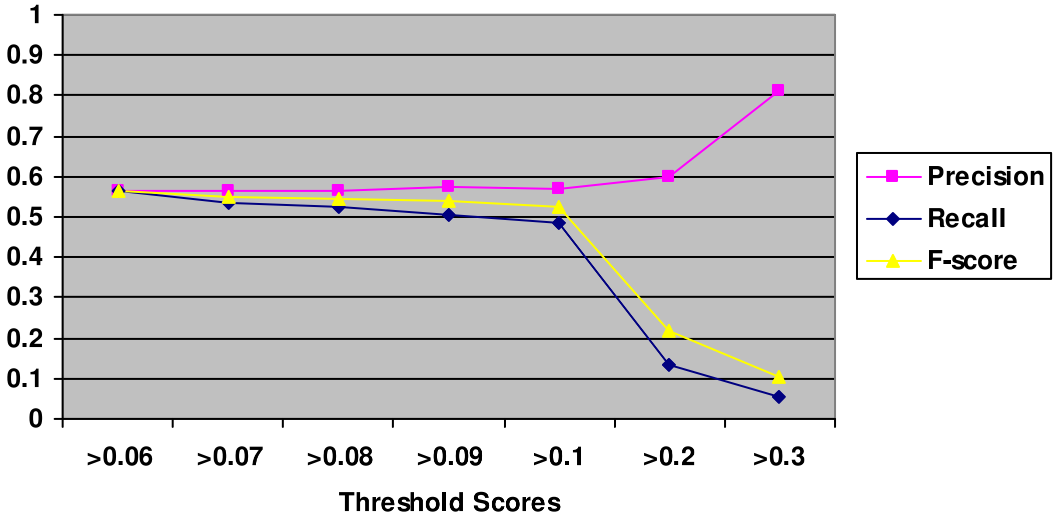


Figure 26: Precision, recall and F-score for adapted linked chain dependency patterns in score-thresholding measure using NMI

Similar to the surface-based approach (Section 3.1.6.4); in the dependency-based approach the score-thresholding measure achieves higher precision than the rank-thresholding. Applications such as MCQ generation, as mentioned earlier, rely on high precision so the score-thresholding method gives us the opportunity to attain higher precision but low recall.

3.3 Comparison between Surface-based and Dependency-based Approaches

In section 3.1, we have discussed different surface type patterns (e.g. untagged word patterns, PoS-tagged word patterns and verb-centred patterns) with and without prepositions and as later the experimental results revealed that the verb-centred pattern type along with prepositions performed better than compared to other pattern types and moreover inclusion of prepositions provide useful insight into extracted semantic relations. We employed different ranking methods and found that CHI and NMI are the best performing ranking methods. CHI is the best performing ranking method in terms of precision scores but recall scores are very low (Figure 18) while using NMI we are able to attain much better recall scores (Figure 19). Moreover, the score-thresholding measure performs better than the rank-thresholding. In Section 3.2, we explored the dependency-based pattern approach and there too we found that overall CHI (Figure 25) and NMI (Figure 26) are the best performing ranking methods while the score-thresholding ranking measure outperforms the rank-thresholding.

In this section, we compare the precision scores obtained by using the best performing ranking methods (NMI and CHI) for the dependency-based patterns with the surface-based verb-centred patterns along with prepositions for the GENIA corpus. Figure 27 shows the comparison of precision scores obtained using NMI ranking method for GENIA corpus between the dependency-based patterns and the surface-based verb-centred patterns along with prepositions.

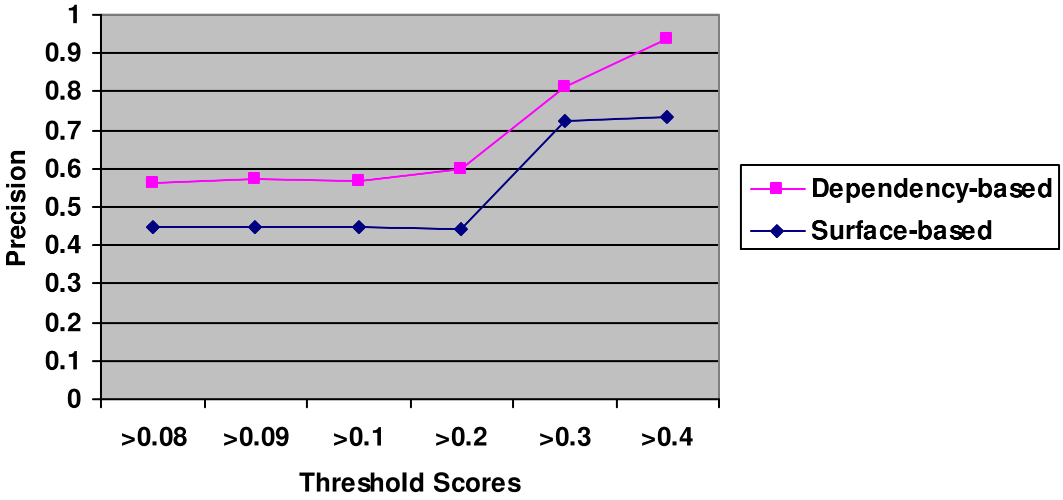


Figure 27: Comparison of precision scores using NMI for GENIA corpus between dependency-based and verb-centred surface-based patterns

Figure 27 shows that the NMI ranking method in dependency-based patterns is able to achieve higher precision scores compare with the NMI ranking method in surface-based verb-centred patterns while Figure 28 shows the same comparison but using CHI ranking method.

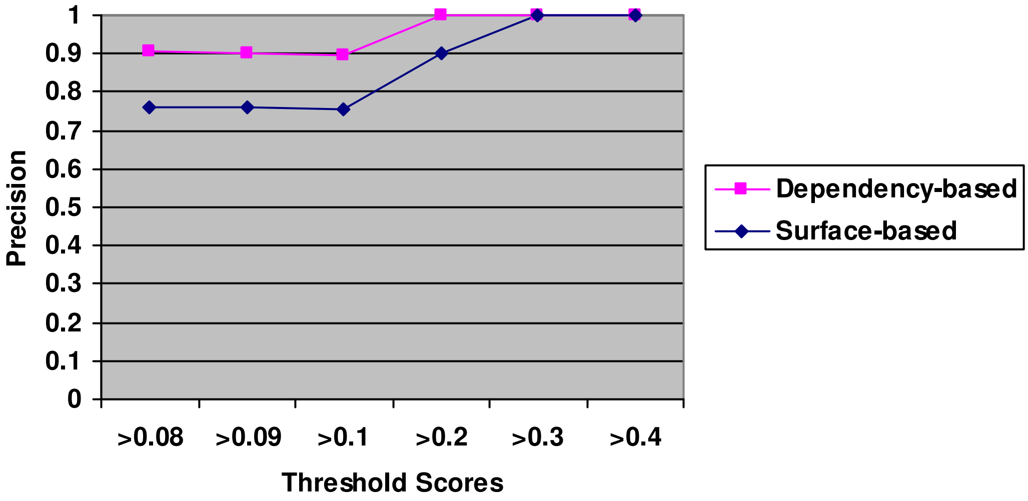


Figure 28: Comparison of precision scores using CHI for GENIA corpus between dependency-based and verb-centred surface-based patterns

Figure 28 also shows that precision scores attained by the dependency-based approach are higher than the scores attained by the surface-based approach.

Overall, the results achieved from Figure 27 and 28 revealed that the dependency-based patterns outperform the best performing surface-based pattern type (verb-centred along with prepositions) in terms of precision scores.

Moreover, the dependency-based approach provided more coverage compared to the surface-based approach. The dependency-based approach enabled us to extract semantic relations that the surface-based approach was unable to extract as it abstract away from different surface realisations of semantic relations. The surface-based approach was able to extract much more effectively those semantic relations that involved PROTEIN and DNA named entities but it was unable to extract a few semantic relations that involved the following named entities (CELL_LINE, CELL_TYPE and RNA) while the dependency-based approach was able to extract these effectively. For example:

[V/express] (subj[CELL_LINE] + obj[RNA])
[V/activate] (p[CELL_LINE] + p[CELL_LINE])
[V/show] (subj[CELL_TYPE] + obj[expression] + prep[of] + P[RNA])
[V/enhance] (a[RNA] + obj[transcription] + prep[in] + p[CELL_LINE])
[V/inhibit] (a[RNA] + obj[transcription] + prep[in] + p[CELL_LINE])
[V/mediate] (obj[transcription] + prep[of] + p[DNA] + prep[in] + p[CELL_LINE])

Our detailed analysis has revealed that the dependency-based approach is much more effective in extracting semantic relations than the surface-based approach.

3.4 Summary

In this chapter, we have presented two unsupervised approaches (surface-based and dependency-based) for Relation Extraction from the biomedical domain. In the surface-based approach, we experimented with three different surface-based approaches and showed that PoS-based and verb-centred patterns achieve higher precision compared to untagged word patterns while in the dependency-based approach we employed an adapted version of a linked chain patterns model to extract the patterns from dependency trees. We explored different ranking methods and found that in the surface-based approach and dependency-based approach the CHI ranking method obtained higher precision than the other ranking methods while NMI is the second best ranking method. In the dependency-based approach we found that we are able to achieve good results if a biomedical corpus is first adapted and then dependency patterns are extracted from it. Moreover, we found that there is no statistical significant difference between IG and IGR, LL and CHI ranking methods in both approaches. We employed two different techniques: the rank-thresholding measure and the score-thresholding measure and found that the score-thresholding measure performs better than the rank-thresholding measure. Moreover, corpus homogeneity and similarity scores revealed that the use of the Web as a corpus is still unable to ensure the same level of topic relevance as achieved in manually compiled corpora. At the end of this chapter, we compared the dependency-based approach with the best performing surface-based approach and found that the dependency-based approach achieves better results than to the best performing surface-based approach.

Chapter 4: Questions and Distractors Generation

In this chapter, we will look at the way extracted patterns (i.e. semantic relations) are transformed into questions automatically. First, we will discuss the approach employed to transform extracted surface-based patterns into questions and then the approach used to transform extracted dependency-based patterns into questions. At the end of this chapter, we will elaborate on the process of automatically generating distractors for each question using a distributional similarity measure.

4.1 Question Generation

Automatic question generation is an important and emerging area of research in NLP. The automatic question generation has the potential to be employed in various areas such as intelligent tutoring systems, dialogue systems (Walker et al., 2001) and educational technologies (Graesser et al., 2005). In automatic question generation it is not only important to ask questions which are grammatically correct but also that the generated questions are asking about important concepts described in a given text (Vanderwende, 2008). Moreover, it is also important to automatically generate questions that stimulate learning process among the learners. Recent workshops in Question Generation Task and Evaluation[34] are trying to define a shared task for question generation. In 2010, Question Generation Shared Task and Evaluation Challenge (QGSTEC[35], 2010) focused on evaluating the generation of questions from paragraphs and the generation of questions from sentences.

It is well-known that generating/asking good questions is a complicated task (Graeseer and Person, 1994). Vanderwende (2007, 2008) emphasised the need of generating important questions from a given text. *Ruminator* (Ureel et al., 2005) is a computer system which generates questions from simplified input sentences but this

[34] http://www.questiongeneration.org/
[35] http://www.questiongeneration.org/QGSTEC2010

system relies heavily on simplified input sentences and it does produce quite a large number of obvious or easy questions. Due to this the quality of the generated question is not particularly good and moreover the generated questions are not informative enough. Another question generation system presented by Schwartz et al. (2004) generates questions in order to help the learning process. This system depends on the summarisation as a pre-processing step for the identification of important questions in a given text. The authors noted that question selections created by the system can be difficult to process.

Gates (2008) presented an approach that could automatically generate fact-based reading comprehension questions by using a look-back strategy i.e. re-reading the text to find the answer of a given question. The system presented in this paper makes use of several existing NLP resources i.e. BBN's IdentiFinder (Bikel et al., 1999) for recognising named entities and specific Prop-Bank (Palmer et al., 2005) semantic arguments (e.g. ARG0, ARG1) using ASSERT (Pradhan et al., 2005). The system uses CBC4Kids corpus (news texts for children) and produces a reading passage along with 5 randomly selected questions and clickable answers in the text. The system measures the accuracy of reading comprehension questions in terms of grammaticality, semantic correctness and practicality of the questions produced from the text. The system was able to generate 81% of acceptable questions from reading comprehensions. The drawback of this system is that most of the questions are quite obvious and too easy to answer.

Chen et al. (2009) presented an approach to generate self-questioning instructions automatically from any given informational text, specially focusing on children's text (children's in grades 1-3). Previous work (Mostow and Chen, 2009) automatically generated self-questioning instructions from narrative text by first generating questions from the text and then augmenting the questions into strategy questions. Narrative text focuses on characters, their behaviours and their mental states (e.g. happy, sad, think, regret) while informational text places emphasis on descriptions and rationalisations of a certain objective phenomena. Due to the different nature of narrative text and informational text the same approach cannot be applied to both of them. The informational text does not contain many mental states so the system has to make use of discourse markers which indicate causal relationships (conditional and temporal contexts such as if, after), modality (i.e. possibility and necessity) and inference rules to generate questions from informational text. The system evaluated the generated questions in terms of their grammatical correctness and how the generated questions made sense in the context of the text. From 444 total sentences in test corpus, the system generated 180 questions in total, 15 questions about conditional contexts (86.7% acceptable), 88 questions about temporal information (65.9% acceptable) and 77 questions about modality (87.0% acceptable).

Kalady et al. (2010) presented an approach to automatically generated questions based on syntactic and keyword modelling. Their approach mainly relied on parse tree manipulation, named entity recognition and Up-keys (significant phrases in a document) to automatically generate factoid and definitional questions from input documents. The factoid questions are generated from a single sentence and are very simple (e.g. yes/no questions and wh-questions from the subject, object, adverbials and prepositional phrases in the sentence). The process of generating definitional questions is quite different as compared to factoid questions as they have descriptive answers and they used the concept of Up-keys that are keywords relating to the input document (Das and Elikkottil, 2010). The authors of this paper only evaluated the factoid-based questions by preparing a gold-standard of questions from a set of documents and comparing the automatically generated questions with them. They reported the results in terms of precision, recall and F-score and their system achieved a precision score of 0.46, recall 0.68 and F-score of 0.55. The main drawback of this approach is its inability to handle lengthy and complex sentences, as well as the fact that the automatically generated questions are very simple and easy to answer.

It still remains a great challenge in the field of NLP to decide which part of the text is important in a given text as identification of key concepts present in a text is a critical sub task during automatic question generation (Nielsen, 2008). Moreover, it is also important for the automatically generated questions to be syntactically and semantically well-formed.

4.2 Our Approach

Our research enables us to generate questions regarding the important concepts present in a domain. This is done by relying on the unsupervised Relation Extraction approach; extracted semantic relations allow us to identify key information in a sentence. In Chapter 3, we extracted important semantic relations present in a domain in the form of patterns and in this chapter we will describe our approach to automatically transform those extracted semantic relations (patterns) into questions. The automatically generated questions by our approach are more effective as it automatically generates questions from important concepts present in the given domain by relying on the semantic relations. Our approach for the automatic generation of questions depends upon accurate output of the named entity tagger and the parser.

4.2.1 Surface-based Patterns

In order to automatically generate questions from surface-based patterns, we first assume that the user has supplied a set of documents on which students will be tested. We will refer to this set of documents as “evaluation corpus” (e.g. in this research, we used a small subset of GENIA EVENT Annotation corpus as an evaluation corpus). In Chapter 3, we have extracted a set of relevance-ranked semantic patterns from the GENIA corpus. As we found that NMI and CHI ranking methods are the best performing ranking methods, we select semantic patterns attaining higher precision/ higher F-score at certain score thresholds using the score-thresholding measure. As in our surface-based approach semantic patterns always start and end with a named entity (see Section 3.1), so we extracted surface-based semantic patterns from the evaluation corpus and try to match these patterns with the semantic patterns learned from the GENIA corpus and when a match is found we extract the whole sentence from the evaluation corpus and then automatically transform the extracted pattern into a question by using certain set of rules (Table 14). This whole automatic question generation process can be illustrated by the following example:

Pattern: DNA contain_v DNA

Step 1: Identify instantiations of a pattern in the evaluation corpus, this involves finding the template (in the above example, the verb 'contain') and the slot filler (two specifics DNA's in the above example). We then have the aforementioned pattern being matched in the evaluation corpus and the relevant sentence is extracted form it.

Thus, the gamma 3 ECS is an inducible promoter containing cis elements that critically mediate CD40L and IL-4-triggered transcriptional activation of the human C gamma 3 gene.

Step 2: The part of the extracted sentence that contains template together with slot fillers is tagged by <QP> and </QP> tags as shown below:

Thus, the <DNA> gamma 3 ECS </DNA> is an ***<QP>*** *<DNA> inducible promoter </DNA> containing <DNA> cis elements </DNA>* ***</QP>*** *that critically mediate <protein> CD40L </protein> and IL-4-triggered transcriptional activation of the <DNA> human C gamma 3 gene </DNA>.*

Step 3: In this step, we extract semantic tags and actual names from the extracted sentence by employing Machinese parser (Tapanainen and Järvinen, 1997). After parsing, the extracted semantic pattern is transformed into the following question:

Which DNA contains cis elements?

As mentioned earlier, our surface-based patterns consisted of two named entities, one at the start and the other at the end of a pattern along with content words and prepositions, so during the automatic questions generation process from various forms of extracted patterns, we develop a certain set of rules (Table 14) based on semantic classes (Named Entities) and part-of-speech (PoS) information present in a pattern. We employ verb-centred patterns along with prepositions for question generation as the presence of a verb between two NEs does generally represent a meaningful semantic relation between them. During evaluation of different types of patterns in Chapter 3, we also found that verb-centred patterns along with prepositions achieve good results in terms of precision, recall and F-score as compared to the untagged word patterns and the PoS-based word patterns. During the automatic generation of questions, we also employed a list of irregular verbs in order to produce past participle form of irregular verbs. Table 14 contains few of the examples of patterns and their respective automatically generated questions. Here SC represents the Semantic Class (e.g. Named Entities). All these rules are domain-independent and only rely on the presence of semantic classes and PoS information between these semantic classes.

Patterns	Questions Examples
SC1 verb SC2 DNA contain_v DNA	Which DNA contains cis elements? Which DNA is contained by inducible promoter?
SC1 verb preposition SC2 CELL_TYPE culture_v with_i PROTEIN	Which cell_type is cultured with IL-4?
SC1 verb adjective SC2 CELL_TYPE express_v several_j PROTEIN	Which cell_type expresses several low molecular weight transmembrane adaptor proteins?
SC1 verb verb SC2 CELL_TYPE exhibit_v enhance_v PROTEIN	Which cell_type exhibits enhance IL-2?
SC1 adverb verb SC2 PROTEIN efficiently_a activate_v DNA	Which DNA is efficiently activated by Oct2?
SC1 verb preposition SC2 PROTEIN bind_v to_t DNA	Which protein binds to ribosomal protein gene promoters?
SC1 verb noun preposition SC2 CELL_LINE confirm_v importance_n of_i PROTEIN	Which cell_line confirms importance of NF-kappa B?
SC1 verb preposition adjective SC2 CELL_TYPE derive_v from_i adherent_j CELL_TYPE	Which cell_type derives from adherent PBMC?
SC1 verb preposition noun preposition SC2 CELL_TYPE result_v in_i activation_n of_i PROTEIN	Which cell_type results in activation of TNF-alpha?
SC1 adverb verb noun preposition SC2 CELL_LINE specifically_a induce_v transcription_n from_i DNA	Which cell_line specifically induces transcription from interleukin-2 enhancer?

Table 14: Examples of extracted patterns along with automatically generated questions

The quality of automatically generated questions in terms of their readability, relevance and acceptance will be evaluated in chapter 5.

4.2.2 Dependency-based Patterns

In a similar way to surface-based patterns approach, we match a learned relevance-ranked dependency-based pattern (GENIA corpus) with a dependency-based pattern of evaluation corpus and the relative sentence is then extracted from the evaluation corpus. The extracted sentence is then automatically transformed into question. The automatic question generation process can be explained by the following example:

Consider the following pattern expressing a semantic relation between two types of proteins:

[V/encode] (subj[DNA] + obj[PROTEIN])

This pattern is matched with the following sentence, which contains its instantiation:

This structural similarity suggests that the pAT 133 gene encodes a transcription factor with a specific biological function.

Our dependency-based patterns always include a main verb, so in order to automatically generate questions we traverse the whole dependency tree of the extracted sentence and extract all of the words which rely on the main verb present in the dependency parse of a sentence.

So from aforementioned sentence, we extracted part from the sentence based on the presence of the main verb from the dependency pattern. The part of the sentence is then transformed into the question by selecting the subtree of the parse bounded by the two named entities present in the dependency pattern. Figure 29 shows the dependency parse of the aforementioned sentence.

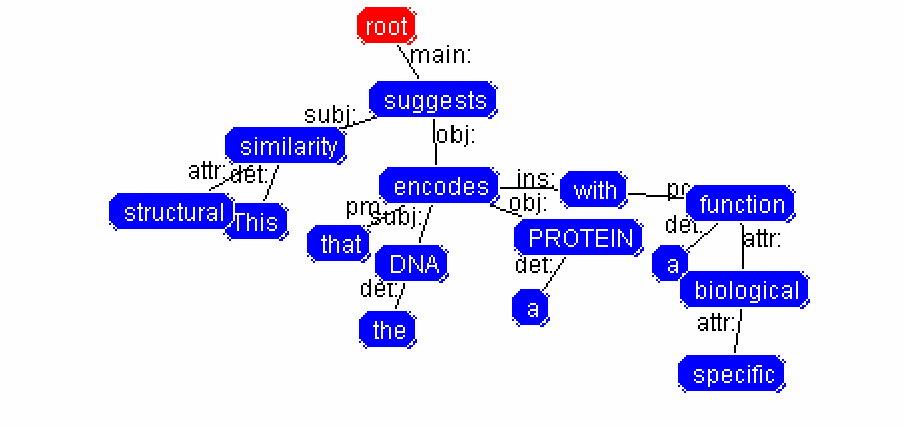


Figure 29: Automatic question generation from dependency tree

From the dependency parse in Figure 29 the following question is automatically generated by traversing the whole dependency tree of the sentence and extracting all of the words that depend on the main verb present in the dependency parse of the sentence:

Which DNA encodes a transcription factor with a specific biological function?

Similar to surface-based questions, the quality of automatically generated dependency-based questions will be evaluated in chapter 5.

In both surface-based and dependency-based approaches, we are able to automatically generate only one type of questions (Which questions) regarding named entities present in a semantic relation. Our approach is not capable of automatically generating different types of questions (e.g. Why, How and What questions), and in order to do that one has to look at various NLG techniques. This would be beyond the scope of this thesis.

4.3 Distractors Generation

Distractors play a vital role in a multiple-choice question as good quality distractors ensure a credible development of the learners' knowledge. The automatic generation of plausible distractors is a very important task in the automatic generation of MCQs. During the process of automatic generation of distractors, the purpose is to find words which are semantically similar to the correct answer but incorrect in the given context.

Goodrich (1977) analysed the potency and discrimination power of manually generated distractors. Previous approaches used different methods in order to automatically generate distractors. Mitkov et al. (2006) used several WordNet-based semantic similarity measures such as the Lesk algorithm (Lesk, 1986), the Jiang and Conrath measure (Jiang and Conrath, 1997), the Lin measure (Lin, 1997) and the Leacock-Chodorow measure (Leacock and Chodorow, 1998) to automatically generate distractors. Most of the previous approaches (e.g. Brown et al., 2005; Sumita et al., 2005 and Hoshino and Nakagawa, 2007) have focused on second language learning acquisition (i.e. grammar and vocabulary). In these approaches distractors are generally generated by employing WordNet, a machine-readable thesaurus or in-house thesauri to retrieve similar words (synonyms, antonyms, hypernyms, hyponyms etc.). Pino et al. (2008) used WordNet to measure semantic similarity while Papasalouros et al. (2008) used domain ontologies built manually by domain experts to automatically generate distractors. Smith et al. (2009) used distributional information from the corpus. Mitkov et al. (2009) argued in their work that semantic similarity measures appear to be a more logical way of automatically generating distractors. They carried their experiments using various semantic similarity measures and found that there is no statistically significant difference between them. Mitkov et al. (2009) used both WordNet and corpora for the automatic generation of distractors. Pino and Eskenazi (2009) presented an automatic approach to generate morphological distractors during cloze questions for English vocabulary learning. In morphological distractors, the distractor is a morphological variant of the correct answer. For example if the correct answer is "interested" then the distractor can be "interesting". In morphological distractors several variant types were generated such as adding –ing or –ed to a verb, -s to a noun, -er or –est to an adjective. Aldabe and Maritxalar (2010) presented a corpus-based approach for the automatic generation of distractors in the Basque language. Their approach made use of semantic similarity measures and ontologies in the process of automatically generating distractors. They used Latent Semantic Analysis (LSA) to compute the context-words similarity.

In order to generate distractors, our approach relies on distributional similarity measures. Distributional similarity is based on the distributional hypothesis which states that words occurring in similar contexts tend to have similar meanings (Harris, 1954; Firth, 1957 and Harshman, 1970). In their work, Mitkov et al. (2006) suggested the usefulness of distributional similarity measures in order to automatically generate plausible distractors. Previous researches have mentioned different levels of context e.g. context of a word in the document in which it occurs, an n-gram, a bag of words on either side or the words with which it has some grammatical dependency.

Distributional similarity is a useful measure and is used in many NLP applications such as language modelling, word classification (Turney and Litman, 2003), query expansion in IR (Cao, et al., 2008), automatic thesaurus generation (e.g. Grefenstette, 1994; Hatzivassiloglou, 1996; Lin, 1998 and Caraballo, 1999), word sense disambiguation (Yuret and Yatbaz, 2010), fact extraction (Paşca et al., 2006), semantic role labelling (Erk, 2007) and textual advertising (Chang et al., 2009). We prefer to use distributional similarity measures in order to automatically generate distractors compared to other taxonomic similarity measures (such as WordNet) as they require having a detailed manually compiled ontology or a resource containing high quality definitions of all possible terms. Another drawback of these taxonomic similarity measures is their limited coverage as they require all candidate named entities and terms found in the instructional material to be recorded in the ontology which itself is a time-consuming and labour-intensive task. Once created, updating ontology is again an expansive and time-consuming task. Moreover, in these manually build lexical resources matching the measure to the resource is a research problem itself as highlighted by Weeds (2003).

Distributional similarity allows us to alleviate the problem of data sparseness by estimating the probabilities of unseen co-occurrences of words from the probabilities of seen co-occurrences of similar words. Moreover, the distributional similarity measure allows us to automatically generate semantically close distractors that are more plausible and better in distinguishing confident test takers from uncertain ones. In distributional similarity similar named entities are generally computed by comparing co-occurrence vectors between all named entities (Sarmento et al., 2007). The advantage of using distributional similarity is that it is corpus-driven compared to manually created lexical resources (Grefenstette, 1994). In order to estimate word co-occurrence probabilities various distributional similarity measures have been proposed (e.g., the L1 Norm, the Euclidean Distance, the Cosine Metric (Salton and McGill, 1983), Jaccard's Coefficient (Frakes and Baeza-Yates, 1992), the Dice Coefficient (Frakes and Baeza-Yates,1992), the Kullback-Leibler Divergence (Cover and Thomas, 1991) and the Jenson-Shannon Divergence (Rao, 1982). Dagan, 2000; Weeds, 2003; Mohammad and Hirst (2005) have presented a detailed review of various distributional similarity measures.

The best distributional similarity measure will be the one which returns the most plausible neighbours in the context of a particular application and thus leads to the best performance in that application. A few/several distributional similarity measures such as Euclidean distance, the cosine and the L1 distance treated the distributions as vectors and made use of geometrically motivated functions to measure distributional similarity. Lee (2001) presented a detailed comparison among various distributional similarity measures. Distributional similarity has also been used in the area of IE. Lin and Pantel (2001) used it to show that patterns which occur with similar pairs tend to have similar meanings. Turney et al. (2003) further showed that pairs of words that co-occur in similar patterns tend to have similar semantic relations.

The distributional hypothesis relies on availability of a large corpus, and is vulnerable to the inevitable data sparseness: reliable estimates of semantic similarity cannot be obtained for infrequent words in the corpus. The availability of a large corpus enables us to examine the context in which words appear and then calculate the similarity between various context distributions.

4.3.1 Our Approach

In order to produce distractors from corpus, we carried out linguistic processing using GENIA tagger. GENIA tagger provides us with tokenised text along with the part-of-speech (PoS) information. In order to handle the data sparseness issue, we build a pool of various biomedical corpora including GENIA, GENIA EVENT, BioInfer[36], YPD (Hodges et al., 1999), Yapex[37], MIPS[38], WEB[39] corpus and BioMed[40] corpus in order to generate distractors from these corpora. After linguistic processing, we build a frequency matrix which involves the scanning of sequential semantic classes (Named Entities) along with a notional word (Noun, Verb, Adverb and Adjective) in the corpora and record their frequencies in a database. In this way, we are able to construct distributional models of all candidate named entities found in the text. Once accurate and informative contextual representation of each semantic class has been extracted along with their frequencies, semantic classes are compared using the distributional hypothesis that similar words appear in similar context. The distractors to a given correct answer are then automatically generated by measuring it similarity to entire candidate named entities. At the end, we select the top 4 similar candidate named entities as the distractors.

Table 15 shows some examples of correct answers and distractors automatically generated by our approach. Our aim is to automatically generate plausible distractors, so if the correct answer is a protein then our approach automatically generates all protein distractors that are involved in similar processes or belong to the same biological category.

[36] http://mars.cs.utu.fi/BioInfer/
[37] http://www.sics.se/humle/projects/prothalt/#data
[38] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC146421/
[39] http://www.ncbi.nlm.nih.gov/
[40] http://www.biomedcentral.com/info/about/datamining/

Correct Answer	**Distractors**			
K562 cells	M1 cells	Yin-Yang 1	Alpha-tubulin	NGF
STAT1	JAK3	NF-kappa B	transcription factor	STAT3
CD40	IL-2	IL-4	T lymphocytes	TCR
monocytes	IFN-gamma	IL-2	NF-kappa B	IL-4
LMP1	HIV-1 Tat	T lymphocytes	NF-kappa B-mediated gene	Fas ligand
ETS transcription factors	beta-promoter	Gammac basal promoter	Human alpha-globin promoter	transgenic thymocytes

Table 15: Examples of automatically generated distractors

In our research, we used grammatical relation data to model context. The use of grammatical relation data to model context in not new as Harris (1968) stated that: "The meaning of entities and the meaning of grammatical relation among them, is related to the restriction of combinations of these entities relative to other entities." We used *Jensen-Shannon divergence* (Rao, 1983 and Lin, J., 1991) also known as *information radius* in order to measure the distributional similarity between two context vectors (i.e. named entities). It is a popular distributional similarity measure based on a smoothed version of Kullback-Leibler's divergence measure (Kullback and Leibler, 1951; Cover and Thomas, 1991; Pereira et al., 1993) and has been frequently employed in word clustering and nearest neighbour techniques (e.g. Dagan et al., 1999; Lapata et al., 2001; Dhilon et al., 2002). The Kullback-Leibler divergence or relative entropy is an asymmetric measure which is employed in order to estimate the similarity between two probability mass functions. Cover and Thomas (1991) defined the relative entropy $(D(p \| q))$ between two distributions p and q as "the inefficiency of assuming that the distribution is q when the true distribution is p". So the relative entropy is:

$$D(p \| q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

Relative entropy will be equal to zero if the two distributions are equal.

Jensen-Shannon divergence is a symmetric measure and is a popular alternative to the Kullback-Leibler divergence measure. Dagan et al. (1999) defined it as "the average of Kullback-Leibler divergence of each of the two distributions to their average distribution".

$$dist_{JS}(p,q) = \frac{1}{2}\left[D(p \,\|\, \frac{p+q}{2}) + D(q \,\|\, \frac{p+q}{2}) \right]$$

Dagan et al. (1997) performs a comparative study based on various distributional similarity measures and found that Jensen-Shannon consistently performs better than other distributional similarity measures.

In chapter 5, we will evaluate the quality of automatically generated distractors in terms of their readability, relevance to the correct answer and their levels of acceptability.

4.4 Summary

In this chapter, we carried out detailed discussion regarding the approaches used to automatically generate questions from both relation extraction approaches (surface-based and dependency-based). In the surface-based approach questions were automatically generated from sentences matched by extracted surface-based semantic relations by relying on a certain set of rules while in the dependency-based approach the questions were automatically generated by traversing the dependency tree of extracted sentence matched by the dependency-based semantic relation. At the end of this chapter, we discussed our approach for automatically generating distractors by using distributional similarity measures. In chapter 5, we will evaluate the automatically generated questions and distractors in terms of their readability, relevance and acceptability.

Chapter 5: Extrinsic Evaluation

In this chapter, we will discuss the importance of extrinsic/user-centred evaluation in any NLP system, evaluation data used during the extrinsic evaluation of both MCQ systems (surface-based and dependency-based) along with the criteria used for the extrinsic evaluation of both systems. We will also elaborate on the results obtained for each MCQ system using the evaluation criteria and compare the evaluation results of both MCQ systems. We involved biomedical experts to extrinsically evaluate both of the systems according to pre-specified evaluation criteria. At the end of this chapter, we will measure the agreement between the two evaluators by employing Kappa statistics.

5.1 Overview

The real application users have a vital role to play in the extrinsic or user-centred evaluation process. The involvement of real users in the evaluation process may vary depending upon the nature of application. According to Paroubek et al. (2007), the user-centred evaluation is a paradigm in which the goal is to analyse the utilisation of the NLP application and its various functionalities by the users in their environment. The user-centred evaluation is quite frequently employed by the Information Retrieval (IR), Machine Translation (MT), Natural Language Generation (NLG) and Automatic Summarisation research community (Hirschman and Mani, 2003; Reiter et al., 2005; Paroubek et al., 2007). In intrinsic evaluation, output produced by the system is compared against the gold-standard (the output produced by humans manually before the evaluation). Precision, recall and F-score are the most frequently used evaluation metrics during the intrinsic/automatic evaluation. Intrinsic evaluation is the most popular and commonly used evaluation measure depending upon the availability of gold-standard. In many NLP applications intrinsic evaluation is used to evaluate components of the application as we have done during the evaluation of our IE component of both MCQ systems. Extrinsic evaluation is a sort of global evaluation in which the application as a whole is evaluated, just as we will be doing in this chapter.

Evaluation has become an integral part of any NLP system (Hirschman and Mani, 2003). The sole purpose of evaluation is to provide a common ground in order to compare systems and approaches. During the process of system evaluation, it is essential for a system to identify all system elements that can figure as performance factors. Spärck Jones and Galliers (1996) made the following observation regarding the process of evaluation: "Evaluation must be designed to address issues relevant to the specific task domain of the NLP system; therefore, NLP systems operating in different task domains require different evaluation criteria." All the stakeholders (funding organisations, research community and end users) want to know how useful the system is in real-life application and the performance of the system in comparison to others. In recent years, the NLP community has invested a lot of time and effort into the evaluation of NLP systems through the organisation of conferences (e.g., Language Resources and Evaluation Conference (LREC[41])) and many evaluation campaigns such as Message Understanding Conferences (MUC [42]), Document Understanding conferences (DUC[43]) and Text Retrieval Conferences (TREC[44]). Text Analysis Conference (TAC) also has many TAC tracks[45] focused on providing a common evaluation procedure that can improve performance of NLP systems on end-user tasks.

There are several different ways to evaluate NLP systems (Paroubek et al., 2007). In *black-box* evaluation (Palmer and Finin, 1990), the evaluation is mainly concerned with the output of the system and not how the system achieves this output. In *white-box/glass-box* evaluation (Palmer and Finin, 1990) all the system components are assessed in order to find out how the system attains these results. Black-box evaluation is relatively easier as compared with white-box evaluation in terms of time and resources.

[41] http://www.lrec-conf.org/
[42] http://www-nlpir.nist.gov/related_projects/muc/
[43] http://www-nlpir.nist.gov/projects/duc/intro.html
[44] http://trec.nist.gov/
[45] http://www.nist.gov/tac/tracks/index.html

5.2 Our Approach

In Chapter 3, we evaluated the IE component of our systems (surface-based and dependency-based) by using the automatic/gold-standard evaluation. In this chapter, we will evaluate both MCQ systems as a whole in a user-centred fashion. The quality of automatically generated MCQs is generally evaluated by human evaluators. The evaluation used in our approach is mainly concerned with the adequate and appropriate generation of MCQs and as well as the amount of human intervention required. In other words, we want to evaluate our system in terms of its robustness and efficiency.

5.2.1 Evaluation Data

For the purpose of the evaluation, we randomly selected a small subset from GENIA EVENT corpus. We found in Chapter 3 that in both surface-based and dependency-based approaches NMI and CHI are the best performing ranking methods during the unsupervised relation extraction phase. CHI achieved very high precision scores but recall scores are very low (Figure 18, 25) while in NMI (Figure 19, 26) recall scores are relatively higher than CHI (see Appendix C for further details). Due to this reason, during the extrinsic evaluation phase of automatically generated MCQ systems we employ NMI for both approaches (surface-based and dependency-based) as it was the only ranking method that enabled us to achieve a higher F-score for both approaches and can provide a better evaluation result for both MCQ systems in terms of its usability and effectiveness. Similarly in Chapter 3, we found that the score-thresholding measure performed better than the rank-thresholding measure, so we have chosen the score-thresholding measure here. We selected a score-thresholding (score > 0.01) for NMI for both approaches as it gives a maximum F-score for both approaches. For surface-based it gives us an F-score of 54% while in dependency-based the F-score is 65%.

5.2.2 Evaluation Method

The extrinsic evaluations of both MCQ systems (surface-based and dependency-based) follow a similar sort of criteria used by Farzindar and Lapalme (2004) for the evaluation of LetSum (an automatic legal text summariser). In LetSum, extrinsic evaluations were based on legal expert judgement. They have defined a series of specific questions for the judgement, which covers the main topics of the document. If a user is able to answer the questions correctly by only reading the summary, it means that the summary contains all of the necessary information from the source judgement. Extrinsic evaluation can measure from what extent a specific NLP application can benefit from employing a certain method or measure.

Both MCQ systems (surface-based and dependency-based) automatically generated 80 and 52 MCQs respectively from the evaluation dataset for NMI score > 0.01. In order to evaluate quality of the automatically generated MCQs, we follow the following criteria:

Readability of automatically generated questions and distractors is evaluated by asking whether it is clear, rather clear or incomprehensible.

Usefulness of semantic relation: Questions are automatically generated by relying on semantic relations, so it is important to evaluate the usefulness of semantic relations present in a question by asking whether it is clear, rather clear or incomprehensible.

Relevance: automatically generated questions should be relevant to the extracted sentence from which the question is generated automatically; similarly for automatically generated distractors it is also important for them to be relevant to the automatically generated question and its answer. Both automatically generated questions and distractors are evaluated in terms of relevance by asking whether it is very relevant, rather relevant or not relevant.

Acceptability: in order to evaluate the acceptability of automatically generated questions and distractors the evaluators are asked to evaluate them from a scale of 0 to 5 (where 0 means unacceptable and 5 means totally acceptable).

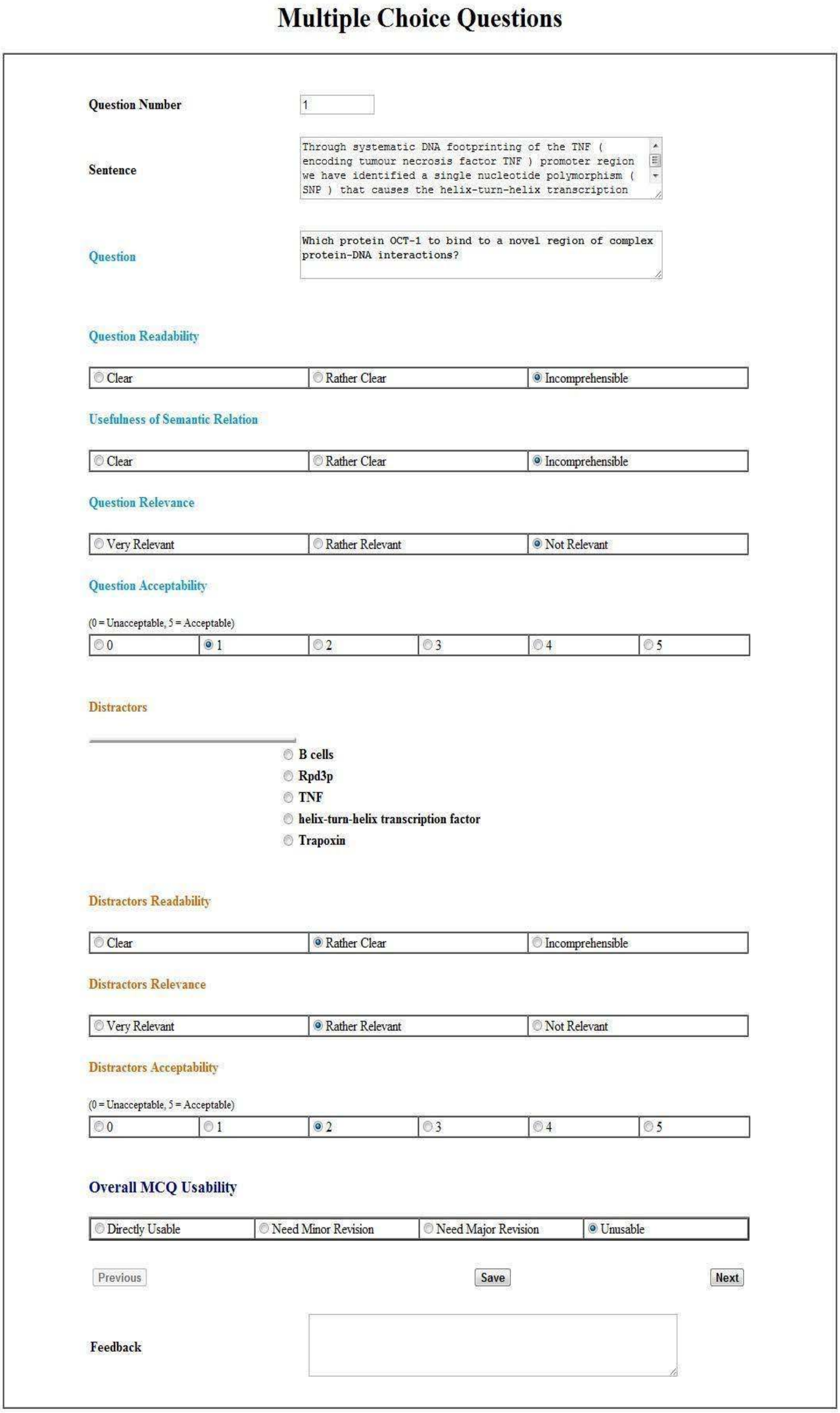
Overall MCQ usability: at the end of this evaluation the evaluators are asked to evaluate the overall usability of automatically generated MCQs by selecting one option from directly usable, needs minor revision, needs major revision or unusable.

Figure 26 shows the screenshot of the interface used during the extrinsic evaluation of both automatically generated MCQs system (Appendix B shows few examples of automatically generated MCQs). The biomedical experts were asked to complete this interface during the extrinsic evaluation of each MCQ.

In the extrinsic evaluation, two biomedical experts (both post-doc) were asked to evaluate both MCQs systems (surface-based and dependency-based) according to the aforementioned criteria. Both evaluators were vastly experienced, one evaluator's[46] main area of research focuses on isolation, characterising and growing stem cells from Keloid and Dupuytren's disease and is currently working at Plastics and Reconstructive Surgery Research while the other biomedical expert[47] is a bio-curator with a PhD in molecular biology and is currently working for the Hugo Gene Nomenclature Committee (HGNC). Both evaluators were asked to give a scoring value for the readability of questions and distractors from 1 (incomprehensible) to 3 (clear), usefulness of semantic relations from 1 (incomprehensible) to 3 (clear), question and distractors relevance from 1 (not relevant) to 3 (very relevant), question and distractors acceptability from 0 (unacceptable) to 5 (acceptable) and overall MCQ usability from 1 (unusable) to 4 (directly usable).

[46] http://www.plasticsurgeryresearch.org/people/PostDocs.html

[47] http://www.genenames.org/about/team



Multiple Choice Questions

Question Number 1

Sentence Through systematic DNA footprinting of the TNF (encoding tumour necrosis factor TNF) promoter region we have identified a single nucleotide polymorphism (SNP) that causes the helix-turn-helix transcription

Question Which protein OCT-1 to bind to a novel region of complex protein-DNA interactions?

Question Readability

○ Clear	○ Rather Clear	● Incomprehensible

Usefulness of Semantic Relation

○ Clear	○ Rather Clear	● Incomprehensible

Question Relevance

○ Very Relevant	○ Rather Relevant	● Not Relevant

Question Acceptability

(0 = Unacceptable, 5 = Acceptable)

○ 0	● 1	○ 2	○ 3	○ 4	○ 5

Distractors

○ **B cells**
○ **Rpd3p**
○ **TNF**
○ **helix-turn-helix transcription factor**
○ **Trapoxin**

Distractors Readability

○ Clear	● Rather Clear	○ Incomprehensible

Distractors Relevance

○ Very Relevant	● Rather Relevant	○ Not Relevant

Distractors Acceptability

(0 = Unacceptable, 5 = Acceptable)

○ 0	○ 1	● 2	○ 3	○ 4	○ 5

Overall MCQ Usability

○ Directly Usable	○ Need Minor Revision	○ Need Major Revision	● Unusable

Feedback

Figure 30: Screenshot of extrinsic evaluation interface

5.2.3 Results

Table 16 shows the results obtained for surface-based and dependency-based MCQ systems where *QR, DR USR, QRelv, DRelv, QA, DA and MCQ Usability* represents *Question Readability, Distractors Readability, Question Relevance, Distractors Relevance, Question Acceptability, Distractors Acceptability and Overall MCQ Usability* respectively.

	QR (1-3)	**DR (1-3)**	**USR (1-3)**	**QRelv (1-3)**	**DRelv (1-3)**	**QA (0-5)**	**DA (0-5)**	**MCQ Usability (1-4)**
Surface-based MCQs System								
Evaluator 1	2.15	2.96	2.14	2.04	2.24	2.53	3.04	2.61
Evaluator 2	1.74	2.29	1.88	1.66	2.10	1.95	3.28	2.11
Average	1.95	2.63	2.01	1.85	2.17	2.24	3.16	2.36
Dependency-based MCQs System								
Evaluator 1	2.42	2.98	2.38	2.37	2.31	3.25	3.73	3.37
Evaluator 2	2.25	2.15	2.46	2.23	2.06	3.27	3.15	2.79
Average	2.34	2.57	2.42	2.30	2.19	3.26	3.44	3.08

Table 16: Evaluation results of surface-based and dependency-based MCQ systems

5.2.4 Comparison

In this section, we performed a comparison between the results of surface-based and dependency-based MCQs systems. For this purpose, we take the average scores of all the categories for each MCQ system and compare them. Figure 31 shows the comparison between the two MCQ systems.

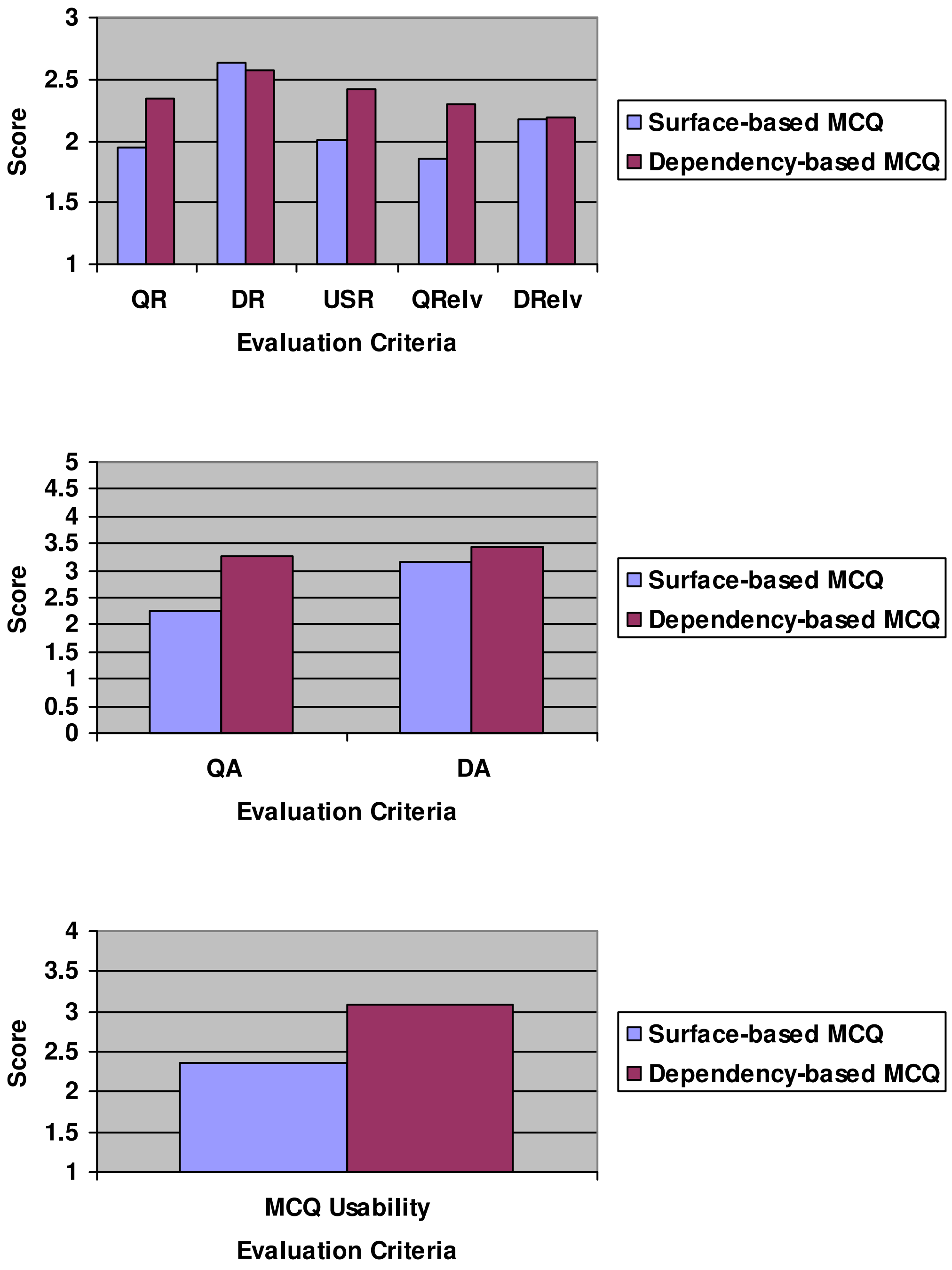


Figure 31: Comparison between surface-based and dependency-based MCQ systems

The results from Figure 31 show that MCQs generated using the dependency-based approach achieve better results during extrinsic evaluation in terms of question readability, usefulness of semantic relation, question and distractors relevance, question and distractors acceptability and overall usability of MCQ. These results are better compared with the extrinsic evaluation results of surface-based MCQs system respectively. In terms of overall MCQ usability, the extrinsic evaluation results show that in surface-based MCQ system 35% of MCQ items were considered directly usable, 30% needed minor revisions and 14% needed major revisions while 21% MCQ items were deemed unusable. In case of dependency-based MCQ system, we found that 65% of MCQ items were considered directly usable, 23% needed minor revisions and 6% needed major revisions while 6% of MCQ items were unusable.

5.2.5 Discussion

We used Kappa statistics (Cohen, 1960) in order to measure the agreement between the two evaluators. Kappa statistics are a quite useful and popular quantitative measure that is used to measure the agreement between evaluators. The Kappa coefficient between evaluators is defined as:

$$K = \frac{P_A - P_E}{1 - P_E}$$

where P_A is the times evaluators agree and P_E is the proportion of times that we would expect the evaluators to agree by chance. K = 1 when there is a complete agreement among the evaluators while K = 0 when there is no agreement. The interpretation of the Kappa score is very important and an example of a commonly used scale is presented in Table 17 (Cohen, 1960).

Kappa Score	Agreement
<0.20	Poor
0.21 – 0.40	Fair
0.41 – 0.60	Moderate
0.61 – 0.80	Good
0.81 – 1.00	Excellent

Table 17: Interpretation of Kappa score

In our extrinsic evaluation, both of the evaluators evaluated both MCQ systems (surface-based and dependency-based) according to the criteria mentioned in the Section 5.1.2. We measured the agreement between the evaluators by using Kappa score which is shown in Table 17.

Evaluation Criteria	**Kappa Score (Surface-based MCQ)**	**Kappa Score (Dependency-based MCQ)**
Question Readability	0.29	0.31
Distractors Readability	0.08	-0.13
Usefulness of Semantic Relation	0.21	0.42
Question Relevance	0.27	0.22
Distractors Relevance	0.29	0.31
Question Acceptability	0.27	0.26
Distractors Acceptability	0.12	0.10
Overall MCQ usability	0.25	0.23

Table 18: Kappa score

The average Kappa score is 0.27 which is fair according to Table 17 but not very high due to various different sub-categories present in the extrinsic evaluation.

We used weighted Kappa (Cohen, 1968) to measure the agreement across major sub-categories in which there is a meaningful difference. For example, in question readability we had three sub-categories: ‘Clear’, ‘Rather Clear’ and ‘Incomprehensible’. In this case we may not care whether one evaluator chooses question readability as ’Clear’ while another evaluator chooses ‘Rather Clear’ in regards to the same question. We might care however if one evaluator chooses question readability as ‘Clear’ while another evaluator chooses question readability for the same question meaning it is recorded as ‘Incomprehensible’. In weighted Kappa, we assigned a score of 1 when both of the evaluators agree while a score of 0.5 is assigned when one evaluator chooses the question readability of a question as

‘Clear’ while the other evaluator chooses it as ‘Rather Clear’. We used a similar sort of criteria during distractors readability, usefulness of semantic relation, question relevance and distractors relevance. In questions and distractors acceptability, we assigned an agreement score of 1 when both evaluators agree completely while a score of 0.5 was assigned when both of the evaluators choose questions and distractors acceptability between ‘0’ and ‘2’. A score of 0.5 was also assigned when both of the evaluators choose questions and distractors acceptability between ‘3’ and ‘5’. In overall MCQ usability, we assigned a score of 1 when both of the evaluators agreed and a score of 0.5 was assigned when one of the evaluator assigned an MCQ as ‘Directly Usable’ while the other evaluators marked the same MCQ as ‘Needs Minor Revision’. An agreement score of 0.5 was assigned when an MCQ was assigned by one of the evaluator as ‘Needs Major Revision’ while the other evaluator marked the same MCQ as ‘Unusable’. Table 19 shows the results obtained using weighted Kappa.

Evaluation Criteria	**Kappa Score (Surface-based MCQ)**	**Kappa Score (Dependency-based MCQ)**
Question Readability	0.44	0.44
Distractors Readability	0.48	0.37
Usefulness of Semantic Relation	0.37	0.51
Question Relevance	0.43	0.42
Distractors Relevance	0.48	0.54
Question Acceptability	0.46	0.45
Distractors Acceptability	0.39	0.39
Overall MCQ usability	0.43	0.41

Table 19: Weighted Kappa score

The results in Table 19 show that the use of weighted Kappa has increased the agreement between the two evaluators from fair to moderate. The agreement between the two evaluators is not very high. Because of this we are not looking at average scores between the two evaluators but instead we analyse the scores assigned by each evaluator separately.

One of the main reasons for not having high agreement score between the two evaluators is that these MCQs are generated from a part of the GENIA EVENT corpus which is very different to an instructional text or teaching material. As mentioned earlier, the GENIA EVENT corpus consists of MEDLINE abstracts so due to that some automatically generated MCQs are ambiguous or lacks context. For example in an MCQ, one evaluator classified the question readability as ‘Clear’ and the same MCQ is classified as ‘Rather Clear’ by the other evaluator due to the lack of context. This can be explained from the following example:

Sentence: *Conversely inhibition of NF-kappaB confers a tenfold increase in glucocorticoid mediated apoptosis establishing that NF-kappaB also functions as an antiapoptotic factor.*

The following question was automatically generated from the aforementioned sentence:
Which protein also functions as an antiapoptotic factor?

According to the feedback of one evaluator this question is ambiguous and needs more context as there are hundreds of apoptotic factors and so there is a possibility of more than one right answer for this question. Similarly *NF-Kappa B* protein refers to a family of several proteins rather than one protein only so context is also important in automatically generating good quality MCQs. Moreover, sometimes the GENIA named entity tagger’s inability to recognise the boundaries of a named entity also resulted in MCQ where the answer of a particular question is partially given in the question. This can be elaborated from the following example:

Sentence: *The B cell-specific nuclear factor OTF-2 positively regulates transcription of the human class II transplantation gene DRA.*

The following question was automatically generated from the aforementioned sentence:

Which protein OTF-2 positively regulates transcription of the human class II transplantation gene DRA?

According to the evaluator's feedback the answer of the question is partially given in the question and the actual question should be:
Which protein positively regulates transcription of the human class II transplantation gene DRA?

But due to the GENIA tagger's inability to recognise some named entity boundaries our system was unable to automatically generate the correct question.

In order to test the significance of the difference between two sets of (surface-based and dependency-based) MCQ systems we used the Chi-Square test, which being a non-parametric statistical test, is suitable as we cannot assume a normal distribution of evaluator scores. In carrying out the test, we compared two sets of scores assigned by one evaluator: the scores assigned to MCT items generated with the surface-based method and those assigned to MCT items generated with the dependency-based method. Table 20 shows the p-values of Chi-Square test obtained from using the evaluation scores provided by the two evaluators.

Evaluation Criteria	**p-values of Chi-Square Test**	
	Evaluator 1	**Evaluator 2**
Question Readability	0.1912	**0.0011**
Distractors Readability	0.5496	0.4249
Usefulness of Semantic Relation	0.2737	**0.0002**
Question Relevance	0.0855	**0.0004**
Distractors Relevance	0.1244	0.7022
Question Acceptability	0.1449	**0.0028**
Distractors Acceptability	0.0715	0.4123
Overall MCQ Usability	**0.0026**	**0.0010**

Table 20: p-values of Chi-Square

In Table 20, where there is a statistical significant difference (at the level of $p < 0.05$), between surface-based and dependency-based MCQ systems, the number is shown in bold. Both evaluators agreed during the extrinsic evaluation that the dependency-based MCQ system is better than the surface-based MCQ system in terms of overall MCQ usability. This has been proved by the p-values of Chi-Square (Table 20). Indeed there is a statistical difference between surface-based and dependency-based MCQ systems in terms of overall MCQ usability. The MCQs generated by the dependency-based system are more usable than the MCQs generated by the surface-based systems.

Our extrinsic evaluation methodology enables us to evaluate automatically generated MCQs in terms of question and distractor readability, usefulness of semantic relation, question and distractor relevance, question and distractor acceptability and overall usability of an MCQ. In 2010, First Question Generation Shared Task Evaluation Challenge (QGSTEC[48]) also used a similar sort of evaluation criteria where they evaluated the automatically generated questions in terms of relevance, question type, syntactic correctness and fluency, ambiguity and variety. Mitkov et al. (2006, see Section 2.1) carried out extrinsic evaluation of their automatically generated MCQ system on a much broader scale by using item response theory (Gronlund, 1982) where they evaluated their MCT items in terms of their difficulty and discrimination. We were unable to carry out such extrinsic evaluation of our MCQ systems due to lack of resources and in the future we would like to explore this evaluation approach for our systems.

5.3 Summary

We already measured the performance of the Information Extraction component of the system using automatic, gold-standard evaluation in terms of precision, recall and F-score in the chapter 3. In this chapter, we used extrinsic evaluation to measure the

[48] http://www.questiongeneration.org/QGSTEC2010

performance of whole MCQ systems based on surface-based and dependency-based semantic patterns. We first elaborated on the importance of evaluation in NLP systems and then evaluation criteria used in this evaluation. Two biomedical experts evaluated both systems on the basis of this pre-defined evaluation criteria and we found that the dependency-based MCQ system performed better than the surface-based MCQ system. Moreover, we used Kappa statistics to measure the agreement between the two evaluators and found that there is a moderate agreement between the two evaluators. We found that there is a statistical significant difference between the overall MCQ usability of dependency-based and surface-based MCQ systems and that MCQs generated by the dependency-based system are more usable than the surface-based MCQ systems.

Chapter 6: Conclusions

This chapter provides a summary of the main contributions of the thesis; presents a review of the whole thesis and outlines directions for future extensions of this work.

6.1 Thesis Contributions

This thesis has presented research in the area of unsupervised relation extraction for e-Learning applications. We mainly focused on the automatic generation of multiple-choice questions (MCQs).

The main aim of this thesis was to use IE methodologies to improve the quality of automatically generated MCQs and to overcome the problems faced by the previous approaches. Most of the previous approaches for automatic generation of MCQs relied on the syntactic structures of sentences to generate questions while different approaches were focused on different methods to automatically generate distractors (Section 2.1). The main drawback of these approaches was that they were unable to automatically generate questions from complex sentences; moreover one of the other problems faced by these approaches was the selection of appropriate sentences for automatic question generation. In contrast, our approach attempts to capture semantic rather than syntactic relations between key terms and named entities in a text. In this way, our approach makes use of semantic relations in order to select the best candidate sentences for question generation.

Our approach consisted of three main phases: in the first phase we used IE methodologies to extract semantic relations and in the second phase we automatically generated questions using these semantic relations. In the third phase distractors were automatically generated using distributional similarity measures. This aim was accomplished through adopted unsupervised relation extraction approaches (surface-based and dependency-based) to extract the important semantic relations from the text. In the surface-based approach, we investigated several surface-based pattern types, while in the dependency-based approach we studied extracted semantic relations based on the dependency tree of a sentence.

We conducted experiments with various information-theoretic and statistical measures to rank candidate semantic patterns by domain relevance as well as meta-ranking (a method that combined multiple pattern-ranking methods). The domain ranking methods were used to select those patterns that capture the most important semantic relations between key notions discussed in domain text. Both surface-based and dependency-based patterns selected in this way were evaluated in terms of precision, recall and F-score. The experimental results revealed that overall in both surface-based and dependency-based approaches Normalised Mutual Information (NMI) and Chi-Square (CHI) were the best performing ranking methods among other methods. Moreover, we studied two different measures to select patterns: score-thresholding measure and rank-thresholding measure and found that the score-thresholding performed better than the rank-thresholding measure.

These extracted semantic relations (surface-based and dependency-based) allowed us to automatically generate better quality questions by focusing on the important concepts present in a given text. In the surface-based approach, questions were automatically generated from semantic relations by using a certain set of rules based on named entities and part-of-speech information present in the surface-based patterns. In the dependency-based approach the questions were automatically generated by traversing the dependency tree of a sentence. As dependency-based patterns always include a main verb, so we traverse the whole dependency tree of the extracted sentence and extract all words which rely on the main verb present in the dependency pattern in order to automatically generate questions.

At the next stage, plausible distractors were automatically generated by using a distributional similarity measure. Distributional similarity is known to adequately model the semantic similarity between lexical expressions and it is used quite frequently in many NLP applications (Section 4.3). There exist several distributional similarity measures and previous studies suggest that Information Radius is one of the best performing distributional similarity measure. Distributional similarity measures are corpus-driven and have a broad coverage compared with the thesaurus-based methods that have a limited coverage. Moreover, we preferred to use distributional similarity measures over taxonomic similarity measures (such as those making use of WordNet) as they require having a detailed manually compiled ontology or a resource containing high quality definitions of all possible terms.

After individual components of the systems were evaluated using intrinsic evaluation (i.e. against gold-standard data), we carried out an extrinsic/user-centred evaluation of the whole integrated MCQ systems. We presented an extrinsic evaluation approach to evaluate the quality of automatically generated MCQs systems. Both MCQs systems were evaluated in terms of question and distractors readability, usefulness of semantic relation, question and distractors relevance, acceptability and the overall usability of automatically generated MCQ. Two domain experts evaluated both the systems according to the aforementioned evaluation criteria and the results revealed that MCQs generated using the dependency-based approach were more usable than compared to the surface-based approach. In this research, we mainly focused on the biomedical domain but the developed methods for pattern extraction, distractors and question generation are quite portable and can easily be extended to other domains too.

6.2 Thesis Review

In this section, we present a brief summary of various chapters of the thesis.

Chapter 1 contained the introduction of the research topic and shed light on the importance of e-Learning and the growing needs of effective and efficient e-Learning applications. The chapter also briefly described the importance of multiple choice questions during assessment and the challenges faced during the automatic generation of multiple choice questions. The chapter also elaborated a set of goals which need to be accomplished for the successful completion of this research.

Chapter 2 presented an overview of the work done so far in the area of automatic generation of multiple choice questions along with the detailed description of drawbacks and achievements of previous automatic multiple choice questions approaches. In this chapter we also defined the concept of Information Extraction (IE), its applications, its subtasks: Named Entity Recognition (NER) and Relation Extraction (RE), evaluation of IE systems, different strategies to perform IE and various machine learning approaches in IE. The chapter also provided an overview of various supervised, semi-supervised and unsupervised IE systems. The chapter also elaborated the importance and growing use of the Web as a corpus and the challenges faced during its use.

Chapter 3 contained the detailed description of the IE phase of this research. In chapter 3, we presented two unsupervised RE approaches (surface-based and dependency-based) that can cover a potentially unrestricted range of semantic relations compared to other RE approaches which can only learn to extract those relations presented in annotated text or seed patterns. In our experiments we employed various information-theoretical and statistical measures to rank extracted semantic patterns and experimental results. This revealed that in both surface-based and dependency-based approaches Normalised Mutual Information and Chi-Square were best performing ranking methods in terms of precision, recall and F-score. In evaluation approaches, we used rank-thresholding and score-thresholding measures and found that the score-thresholding performed better than the rank-thresholding measure. In the surface-based approach, we explored three different pattern types without prepositions and with prepositions. The experimental results divulged that verb-centred surface patterns along with prepositions were the best among the other surface pattern types. We also performed the comparison between the best performing surface-based approach and dependency-based patterns approach and found that the dependency-based approach attained better results than compared to the surface-based approach. Our unsupervised RE approaches were able to achieve high precision scores, which was very important as having high precision scores allowed us to automatically generate good quality MCQs.

Chapter 4 contained the detailed description of how semantic patterns (surface-based and dependency-based) are automatically transformed into good quality questions. Our approach enabled us to identify an important part of text in a given text, which was worth asking a question about by using these extracted semantic relations.

Plausible distractors were automatically generated by using a distributional similarity measure. The reason behind choosing a distributional similarity measure was that it is corpus-based, alleviated the problem of data sparseness and provided good coverage compared to other taxonomic similarity measures that required a detailed manually compiled ontologies and had limited coverage. For the automatic generation of distractors, we collected various biomedical corpora and built a frequency matrix of semantic classes (named entities) along with a notional word in corpora. This enabled our distributional similarity measure to automatically generate distractors (similar words expressions) appearing in similar contexts.

Chapter 5 described an extrinsic evaluation method to evaluate the quality of both MCQs systems (surface-based and dependency-based) in terms of question and distractor readability, usefulness of semantic relation, question and distractor relevance, acceptability and the overall usability of automatically generated MCQ. Two biomedical experts independently evaluated both MCQs systems according to aforementioned evaluation criteria. The results of this evaluation revealed that the quality and usability of MCQs generated by the dependency-based MCQs system were much better than the surface-based MCQs system.

6.3 Future Work

During this research and the development of the automatic generation of MCQ systems, a series of potential future leads have emerged. These remain unaddressed in this thesis due to the unavailability of resources and time restrictions. They are discussed in this section.

One of the major advantages of our approach to the automatic generation of MCQs is its domain-independence and portability. It makes use of unsupervised semantic relation extraction method so that it can easily adaptable for other domains. In the future, we would like to extend our approach in other domains. A further direction of research is to demonstrate its portability to other specialist domains and to study its dependence on the amount and quality of corpora from which IE patterns are learned.

The IE component of our automatically generated MCQs systems is based on the semantic relation extraction assumption that it is between named entities stated in the same sentence when that presence or absence of a relation is independent of the text prior to or succeeding the sentence. It will be interesting to investigate a relation extraction process from multiple sentences rather than a single sentence. Moreover, before the relation extraction process from a given text, it will increase the number of extracted semantic relations and ultimately the quality of automatically generated MCQs, if the given text is first processed by the anaphora and co-reference resolution system which replaces all anaphors with its antecedents and then semantic relations are extracted from the text. In the IE phase, we used Machinese parser during the dependency-based approach. It would be interesting to investigate what kind of impact other parsers such as MINIPAR[49] and Stanford parsers[50] will have in terms of precision, recall and F-score of relation extraction process. The semantic relations can also be useful in other applications such as testing reading comprehension where this IE component can identify important concepts in a given text and show which part of the learning material is vital and worth testing.

The automatic question generation phase may benefit from the use of NLG technology (McIntyre and Lapata, 2009; Barzilay and Lapata, 2005; Reiter and Dale, 2000) to improve the quality and grammaticality of automatically generated questions. Another direction of future work is to improve the quality of automatically generated questions further and use them in intelligent tutoring systems, dialogue systems and game-based learning environments.

In automatic distractor generation, we used a distributional similarity measure for automatic distractor generation which is a corpus-driven approach. The Web, the biggest available corpus to the research community is quite frequently used in many NLP applications today, so it would be interesting to investigate the use of the Web as a source for automatic distractors generation. Wikipedia[51] is another useful resource that can also be employed in automatic distractors generation.

[49] http://webdocs.cs.ualberta.ca/~lindek/minipar.htm

[50] http://www-nlp.stanford.edu/software/lex-parser.shtml

[51] http://en.wikipedia.org/wiki/Main_Page

It would be interesting to carry out the extrinsic evaluation of our MCQ systems on a much broader scale using item response theory (Gronlund, 1982). Mitkov et al. (2006) used this theory during the extrinsic evaluation of their MCQ system in which they have evaluated MCT items in terms of their difficulty and discrimination.

In the future, our approach for automatic generation of MCQs can be personalised to help to address the potential knowledge gaps of individuals. In this way, our approach can provide significant assistance to teachers and instructors during the entire learning process.

Appendix A: Previously Published Work

This appendix provides a brief description of the papers included in this thesis that have been previously published in proceedings of peer-reviewed and well-known international conferences. The papers are extended to address the shortcomings identified after the publication of these papers and are then included in this thesis.

- **Afzal, N.** & Pekar, V. (2009). Unsupervised relation extraction for automatic generation of multiple-choice questions. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP2009)*. Borovets, Bulgaria, pp. 1-5.

  This paper presents unsupervised surface-based relation extraction approach. The findings of this paper are described in the Section 3.1 of the Chapter 3.

- **Afzal, N.**, Mitkov, R. & Farzindar, A. (2011). Unsupervised relation extraction using dependency trees for automatic generation of multiple-choice questions. In *Proceedings of the C. Butz and P. Lingras (Eds.): Canadian Artificial Intelligence, LNAI 6657*. Newfoundland and Labrador, Canada: Springer, Heidelberg, pp. 32-43.

  This paper presents an unsupervised dependency-based relation extraction approach. The findings of this paper are used in the Section 3.2 of the Chapter 3.

Appendix B: Examples of Automatically Generated MCQs

This appendix contains few examples of automatically generated MCQs using dependency-based approach along with the sentences from which the MCQ is automatically generated.

Sentence: PPARalpha activators inhibit cytokine-induced vascular cell adhesion molecule-1 expression in human endothelial cells.

Which protein activators inhibit cytokine-induced vascular cell adhesion molecule-1 expression in human endothelial cells?

- Interleukin-5
- PPARalpha
- cultured human ECs
- lymphoid and myeloid cells
- proinflammatory mediator

Sentence: Taken together these results indicate that STAT1 plays a pivotal role in the differentiation/maturation process of monocytes as an early transcription factor initially activated by adherence and then able to modulate the expression of functional genes such as ICAM-1 and FcgammaRI.

Which protein plays a pivotal role in the differentiation/maturation process of monocytes as an early transcription factor?

- JAK3
- NF-kappa B
- STAT1
- transcription factor
- STAT3

Sentence: We show that TLR2 associates with the high-affinity LPS binding protein membrane CD14 to serve as an LPS receptor complex and that LPS treatment enhances the oligomerization of TLR2.

Which protein associates with the high-affinity LPS binding protein membrane CD14?

- Phosphatidylinositol 3-kinase
- T-cell-specific transcription factor
- TLR2
- eukaryotic transcription factor
- HLA-DM

Sentence: We have found that ISG expression in the monocytic U937 cell line differs from most cell lines previously examined.

Which protein expression in the monocytic U937 cell line differs from most cell lines?

- ISG
- SOCS-1
- beta-like globin cluster
- early growth response-1 gene
- Rel/NF-kappa B

Sentence: We show here that c-Rel binds to kappa B sites as homodimers as well as heterodimers with p50.

Which protein binds to kappa B sites as homodimers as well as heterodimers?

- B cells
- NF-kappa B
- NF-kappa B
- c-Rel
- p65

Sentence: We also present evidence that IL-6 kappa B binding factor II functions as a repressor specific for IL-6 kappa B-related kappa B motifs in lymphoid cells.

Which protein functions as a repressor specific for IL-6 kappa B-related kappa B motifs in lymphoid cells?

- IL-6 kappa B binding factor II
- Translocated hormone/receptor complexes
- positive and negative regulatory factors
- recombinant caspase 3
- p1-79 probes

Sentence: The long terminal repeat (LTR) region of HIV proviral DNA contains binding sites for nuclear factor kappa B (NF-kappa B) and this transcriptional activator appears to regulate HIV activation.

Which DNA region of HIV proviral DNA contains binding sites for nuclear factor kappa B (NF-kappa B)?

- Epstein-Barr viral DNA
- chronically infected T cell line
- long terminal repeat
- transcription factor family
- IL-1alpha gene

Sentence: We report here that the HIV-1-encoded Nef protein inhibits the induction of NF-kappa B DNA-binding activity by T- cell mitogens.

Which protein inhibits the induction of NF-kappa B DNA-binding activity by T-cell mitogens?

- HIV-1-encoded Nef protein
- immediate precursors

- prognostic factor
- reticulocytes
- metastasis-suppressor gene

Sentence: We have found that the p49 (100) DNA binding subunit together with p65 can act in concert with Tat-I to stimulate the expression of HIV-CAT plasmid.

Which protein together with p65 can act in concert with Tat-I?

- HLA DQA1*0201
- human PAX-5 gene
- p49 (100) DNA binding subunit
- raf
- immune system regulatory and effector cells

Appendix C: Result Tables

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Top 100 Ranked Patterns									
IG	0.530	0.009	0.018	0.150	0.003	0.005	0.200	0.003	0.007
IGR	0.560	0.010	0.019	0.150	0.003	0.005	0.200	0.003	0.007
MI	0.330	0.006	0.011	0.030	0.001	0.001	0.080	0.001	0.003
NMI	0.680	0.012	0.023	0.390	0.007	0.013	0.550	0.010	0.019
LL	0.560	0.010	0.019	0.150	0.003	0.005	0.200	0.003	0.007
CHI	**0.740**	0.013	0.025	**0.570**	0.010	0.019	**0.640**	0.011	0.022
Meta	0.740	0.013	0.025	0.480	0.008	0.016	0.540	0.009	0.018
tf-idf	0.660	0.011	0.023	0.380	0.007	0.013	0.530	0.009	0.018
Top 200 Ranked Patterns									
IG	0.560	0.019	0.038	0.210	0.007	0.014	0.200	0.007	0.013
IGR	0.565	0.020	0.038	0.210	0.007	0.014	0.205	0.007	0.014
MI	0.305	0.011	0.020	0.030	0.001	0.002	0.105	0.004	0.007
NMI	0.530	0.018	0.036	0.380	0.013	0.025	0.455	0.016	0.031
LL	0.565	0.020	0.038	0.210	0.007	0.014	0.205	0.007	0.014
CHI	**0.615**	0.021	0.041	**0.465**	0.016	0.031	**0.540**	0.019	0.036
Meta	0.605	0.021	0.041	0.315	0.011	0.021	0.430	0.015	0.029
tf-idf	0.525	0.018	0.035	0.375	0.013	0.025	0.390	0.014	0.026
Top 300 Ranked Patterns									
IG	0.543	0.028	0.054	0.173	0.009	0.017	0.213	0.011	0.021
IGR	0.540	0.028	0.053	0.173	0.009	0.017	0.217	0.011	0.021
MI	0.343	0.018	0.034	0.037	0.002	0.004	0.120	0.006	0.012
NMI	0.540	0.028	0.053	0.320	0.017	0.032	0.400	0.021	0.040
LL	0.540	0.028	0.053	0.173	0.009	0.017	0.217	0.011	0.021
CHI	**0.577**	0.030	0.057	**0.387**	0.020	0.038	**0.483**	0.025	0.048
Meta	0.543	0.028	0.054	0.317	0.016	0.031	0.377	0.020	0.037
tf-idf	0.527	0.027	0.052	0.313	0.016	0.031	0.347	0.018	0.034

Table 1: Rank-thresholding results of untagged word patterns

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Top 100 Ranked Patterns									
IG	0.780	0.014	0.027	0.320	0.006	0.011	0.300	0.005	0.010
IGR	0.790	0.014	0.027	0.320	0.006	0.011	0.300	0.005	0.010
MI	0.430	0.008	0.015	0.030	0.001	0.001	0.080	0.001	0.003
NMI	0.810	0.014	0.028	0.520	0.009	0.018	0.660	0.012	0.023
LL	0.790	0.014	0.027	0.320	0.006	0.011	0.300	0.005	0.010
CHI	**0.900**	0.016	0.031	**0.700**	0.012	0.024	**0.800**	0.014	0.028
Meta	0.860	0.015	0.030	0.390	0.007	0.014	0.460	0.008	0.016
tf-idf	0.800	0.014	0.028	0.420	0.007	0.015	0.520	0.009	0.018
Top 200 Ranked Patterns									
IG	0.755	0.027	0.051	0.380	0.013	0.026	0.415	0.015	0.028
IGR	0.755	0.027	0.051	0.380	0.013	0.026	0.415	0.015	0.028
MI	0.420	0.015	0.029	0.050	0.002	0.003	0.125	0.004	0.009
NMI	0.720	0.025	0.049	0.480	0.017	0.033	0.545	0.019	0.037
LL	0.755	0.027	0.051	0.380	0.013	0.026	0.415	0.015	0.028
CHI	**0.755**	0.027	0.051	**0.565**	0.020	0.038	**0.570**	0.020	0.039
Meta	0.765	0.027	0.052	0.400	0.014	0.027	0.480	0.017	0.033
tf-idf	0.715	0.025	0.049	0.440	0.016	0.030	0.490	0.017	0.033
Top 300 Ranked Patterns									
IG	0.720	0.038	0.072	0.307	0.016	0.031	0.353	0.019	0.035
IGR	0.730	0.039	0.073	0.303	0.016	0.030	0.353	0.019	0.035
MI	0.460	0.024	0.046	0.043	0.002	0.004	0.140	0.007	0.014
NMI	0.707	0.037	0.071	0.410	0.022	0.041	**0.503**	0.027	0.051
LL	0.730	0.039	0.073	0.303	0.016	0.030	0.353	0.019	0.035
CHI	**0.740**	0.039	0.074	**0.423**	0.022	0.043	0.500	0.026	0.050
Meta	0.727	0.038	0.073	0.407	0.021	0.041	0.480	0.025	0.048
tf-idf	0.677	0.036	0.068	0.373	0.020	0.037	0.430	0.023	0.043

Table 2: Rank-thresholding results of PoS-tagged word patterns

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Top 100 Ranked Patterns									
IG	**0.840**	0.021	0.041	0.380	0.009	0.018	0.410	0.010	0.020
IGR	0.840	0.021	0.041	0.380	0.009	0.018	0.410	0.010	0.020
MI	0.380	0.009	0.018	0.060	0.001	0.003	0.100	0.002	0.005
NMI	0.790	0.020	0.038	0.380	0.009	0.018	0.480	0.012	0.023
LL	0.840	0.021	0.041	0.380	0.009	0.018	0.410	0.010	0.020
CHI	0.820	0.020	0.040	**0.540**	0.013	0.026	**0.620**	0.015	0.030
Meta	0.830	0.021	0.040	0.370	0.009	0.018	0.380	0.009	0.018
tf-idf	0.780	0.019	0.038	0.390	0.010	0.019	0.450	0.011	0.022
Top 200 Ranked Patterns									
IG	**0.765**	0.038	0.073	0.335	0.017	0.032	0.410	0.010	0.020
IGR	0.765	0.038	0.073	0.340	0.017	0.032	0.410	0.010	0.020
MI	0.360	0.018	0.034	0.040	0.002	0.004	0.160	0.008	0.015
NMI	0.710	0.035	0.067	0.330	0.016	0.031	0.395	0.020	0.038
LL	0.765	0.038	0.073	0.340	0.017	0.032	0.410	0.010	0.020
CHI	0.735	0.037	0.070	**0.365**	0.018	0.035	**0.465**	0.023	0.044
Meta	0.750	0.037	0.071	0.310	0.015	0.029	0.395	0.020	0.038
tf-idf	0.690	0.034	0.066	0.320	0.016	0.030	0.435	0.022	0.041
Top 300 Ranked Patterns									
IG	**0.770**	0.058	0.107	0.263	0.020	0.037	0.357	0.027	0.050
IGR	0.760	0.057	0.106	0.267	0.020	0.037	0.353	0.026	0.049
MI	0.413	0.031	0.058	0.040	0.003	0.006	0.157	0.012	0.022
NMI	0.603	0.045	0.084	0.247	0.018	0.034	0.330	0.025	0.046
LL	0.757	0.057	0.105	0.260	0.019	0.036	0.353	0.026	0.049
CHI	0.623	0.047	0.087	**0.297**	0.022	0.041	**0.367**	0.027	0.051
Meta	0.667	0.050	0.093	0.277	0.021	0.039	0.327	0.024	0.045
tf-idf	0.597	0.045	0.083	0.283	0.021	0.039	0.337	0.025	0.047

Table 3: Rank-thresholding results of verb-centred word patterns

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Threshold score > 0.01									
IG	0.354	0.461	0.401	0.067	0.064	0.065	0.106	0.143	0.122
IGR	0.348	0.358	0.353	0.060	0.065	0.062	0.110	0.125	0.117
MI	0.354	0.518	0.420	0.025	0.126	0.041	0.082	0.523	0.141
NMI	0.357	0.441	0.395	0.027	0.121	0.044	0.083	0.457	0.140
LL	0.349	0.353	0.351	0.060	0.065	0.062	0.110	0.125	0.117
CHI	0.348	0.230	0.277	0.158	0.047	0.072	0.228	0.064	0.099
Meta	0.353	0.392	0.372	0.026	0.097	0.042	0.083	0.383	0.136
tf-idf	0.332	0.265	0.295	0.059	0.085	0.070	0.100	0.332	0.154
Threshold score > 0.02									
IG	0.355	0.280	0.313	0.078	0.043	0.055	0.121	0.079	0.095
IGR	0.356	0.213	0.266	0.080	0.043	0.056	0.123	0.076	0.094
MI	0.355	0.429	0.388	0.026	0.115	0.043	0.082	0.450	0.139
NMI	0.354	0.384	0.368	0.028	0.108	0.045	0.084	0.404	0.139
LL	0.356	0.213	0.266	0.080	0.043	0.056	0.123	0.076	0.094
CHI	0.342	0.163	0.221	0.282	0.033	0.058	0.326	0.039	0.070
Meta	0.347	0.294	0.318	0.029	0.083	0.043	0.086	0.309	0.135
tf-idf	0.326	0.238	0.275	0.063	0.066	0.064	0.114	0.311	0.167
Threshold score > 0.03									
IG	0.354	0.197	0.253	0.089	0.035	0.051	0.131	0.056	0.078
IGR	0.482	0.064	0.113	0.085	0.033	0.047	0.131	0.056	0.078
MI	0.353	0.392	0.372	0.027	0.105	0.043	0.083	0.413	0.138
NMI	0.348	0.327	0.337	0.028	0.097	0.044	0.084	0.361	0.136
LL	0.482	0.064	0.113	0.085	0.033	0.047	0.130	0.055	0.078
CHI	0.339	0.159	0.216	0.314	0.025	0.047	0.386	0.029	0.054
Meta	0.348	0.258	0.296	0.033	0.073	0.045	0.091	0.253	0.134
tf-idf	0.304	0.201	0.242	0.064	0.053	0.058	0.132	0.251	0.173
Threshold score > 0.04									
IG	0.479	0.058	0.104	0.095	0.025	0.039	0.148	0.044	0.068
IGR	0.511	0.050	0.092	0.099	0.026	0.042	0.149	0.044	0.068
MI	0.350	0.368	0.359	0.026	0.099	0.042	0.083	0.397	0.137
NMI	0.344	0.305	0.323	0.029	0.091	0.044	0.084	0.340	0.135
LL	0.511	0.050	0.092	0.099	0.026	0.042	0.149	0.044	0.068

CHI	0.571	0.029	0.055	0.413	0.019	0.037	0.518	0.023	0.044
Meta	0.346	0.219	0.268	0.043	0.063	0.051	0.104	0.198	0.137
tf-idf	0.324	0.161	0.215	0.075	0.042	0.054	0.160	0.223	0.187
Threshold score > 0.05									
IG	0.515	0.044	0.082	0.102	0.021	0.035	0.153	0.037	0.060
IGR	0.511	0.043	0.079	0.102	0.021	0.035	0.153	0.037	0.060
MI	0.349	0.341	0.345	0.026	0.093	0.041	0.083	0.368	0.135
NMI	0.346	0.290	0.316	0.028	0.086	0.042	0.085	0.330	0.135
LL	0.510	0.043	0.079	0.102	0.021	0.035	0.153	0.037	0.060
CHI	0.585	0.024	0.047	0.462	0.017	0.032	0.543	0.019	0.036
Meta	0.340	0.210	0.260	0.042	0.059	0.049	0.105	0.191	0.136
tf-idf	0.350	0.148	0.208	0.092	0.035	0.051	0.212	0.196	0.204
Threshold score > 0.06									
IG	0.537	0.034	0.063	0.111	0.018	0.031	0.168	0.032	0.053
IGR	0.551	0.032	0.061	0.111	0.018	0.031	0.167	0.031	0.052
MI	0.344	0.319	0.331	0.027	0.089	0.041	0.083	0.350	0.134
NMI	0.342	0.265	0.299	0.029	0.083	0.043	0.086	0.310	0.134
LL	0.551	0.032	0.061	0.111	0.018	0.031	0.167	0.031	0.052
CHI	0.576	0.023	0.044	0.516	0.014	0.027	0.589	0.015	0.030
Meta	0.344	0.171	0.229	0.126	0.047	0.069	0.172	0.080	0.109
tf-idf	0.352	0.115	0.173	0.119	0.029	0.047	0.340	0.130	0.188
Threshold score > 0.07									
IG	0.544	0.029	0.055	0.113	0.016	0.028	0.168	0.027	0.047
IGR	0.537	0.028	0.052	0.113	0.015	0.027	0.169	0.027	0.047
MI	0.344	0.315	0.329	0.026	0.086	0.040	0.082	0.343	0.133
NMI	0.341	0.261	0.295	0.029	0.081	0.042	0.086	0.303	0.134
LL	0.536	0.027	0.052	0.113	0.015	0.027	0.169	0.027	0.047
CHI	0.733	0.015	0.030	0.558	0.012	0.024	0.638	0.013	0.025
Meta	0.341	0.169	0.226	0.134	0.046	0.068	0.173	0.073	0.103
tf-idf	0.360	0.075	0.124	0.144	0.024	0.041	0.434	0.116	0.182
Threshold score > 0.08									
IG	0.538	0.026	0.049	0.129	0.013	0.024	0.170	0.025	0.043
IGR	0.537	0.023	0.044	0.114	0.013	0.023	0.169	0.025	0.043
MI	0.340	0.299	0.318	0.026	0.084	0.040	0.082	0.330	0.132
NMI	0.339	0.254	0.291	0.029	0.079	0.043	0.087	0.296	0.134

LL	0.537	0.023	0.044	0.114	0.013	0.023	0.170	0.025	0.043
CHI	**0.750**	0.014	0.028	**0.569**	0.011	0.021	**0.640**	0.011	0.022
Meta	0.526	0.037	0.069	0.153	0.042	0.066	0.190	0.065	0.096
tf-idf	0.398	0.062	0.107	0.201	0.017	0.031	0.451	0.095	0.157
Threshold score > 0.09									
IG	0.562	0.022	0.042	0.148	0.012	0.022	0.188	0.022	0.039
IGR	0.560	0.022	0.042	0.148	0.012	0.022	0.188	0.022	0.039
MI	0.338	0.293	0.314	0.026	0.081	0.039	0.082	0.325	0.132
NMI	0.342	0.240	0.282	0.029	0.076	0.042	0.088	0.286	0.135
LL	0.560	0.022	0.042	0.144	0.012	0.022	0.188	0.022	0.039
CHI	**0.740**	0.013	0.025	**0.579**	0.010	0.019	**0.687**	0.010	0.020
Meta	0.529	0.035	0.066	0.159	0.040	0.064	0.196	0.061	0.093
tf-idf	0.548	0.033	0.063	0.259	0.014	0.026	0.467	0.072	0.125
Threshold score > 0.1									
IG	0.563	0.021	0.040	0.159	0.011	0.020	0.200	0.018	0.033
IGR	0.564	0.021	0.040	0.192	0.011	0.020	0.200	0.018	0.033
MI	0.341	0.281	0.308	0.026	0.079	0.039	0.083	0.322	0.132
NMI	0.341	0.237	0.280	0.029	0.074	0.041	0.087	0.282	0.134
LL	0.562	0.020	0.040	0.188	0.010	0.019	0.200	0.018	0.033
CHI	**0.806**	0.010	0.020	**0.614**	0.009	0.017	**0.711**	0.009	0.018
Meta	0.524	0.034	0.064	0.153	0.037	0.060	0.255	0.046	0.078
tf-idf	0.589	0.019	0.038	0.294	0.010	0.018	0.491	0.053	0.096
Threshold score > 0.2									
IG	0.667	0.007	0.014	0.164	0.005	0.009	0.203	0.007	0.014
IGR	0.667	0.007	0.014	0.164	0.005	0.009	0.203	0.007	0.014
MI	0.337	0.232	0.275	0.025	0.066	0.036	0.085	0.269	0.129
NMI	0.338	0.159	0.216	0.041	0.054	0.047	0.103	0.178	0.131
LL	0.667	0.007	0.014	0.164	0.005	0.009	0.203	0.007	0.014
CHI	**1.000**	0.004	0.009	**0.697**	0.004	0.008	**0.688**	0.004	0.008
Meta	0.800	0.009	0.018	0.300	0.011	0.022	0.382	0.016	0.031
tf-idf	0.709	0.013	0.025	0.345	0.007	0.013	0.541	0.045	0.084
Threshold score > 0.3									
IG	0.568	0.004	0.009	0.165	0.003	0.006	0.203	0.004	0.008
IGR	0.568	0.004	0.009	0.165	0.003	0.006	0.203	0.004	0.008
MI	0.337	0.213	0.261	0.026	0.059	0.036	0.086	0.244	0.127

NMI	0.550	0.025	0.048	0.138	0.041	0.064	0.177	0.065	0.095
LL	0.568	0.004	0.009	0.165	0.003	0.006	0.203	0.004	0.008
CHI	**1.000**	0.002	0.004	**0.727**	0.003	0.006	**0.875**	0.002	0.005
Meta	1.000	0.004	0.008	0.683	0.005	0.010	0.600	0.006	0.011
tf-idf	0.714	0.008	0.015	0.375	0.004	0.008	0.614	0.022	0.042
Threshold score > 0.4									
IG	0.750	0.002	0.004	0.480	0.002	0.004	0.500	0.003	0.005
IGR	0.733	0.002	0.004	0.480	0.002	0.004	0.500	0.003	0.005
MI	0.329	0.190	0.241	0.027	0.052	0.036	0.088	0.213	0.124
NMI	0.544	0.019	0.038	0.157	0.034	0.056	0.198	0.053	0.084
LL	0.733	0.002	0.004	0.480	0.002	0.004	0.500	0.003	0.005
CHI	**1.000**	0.002	0.003	0.917	0.002	0.004	**1.000**	0.002	0.003
Meta	1.000	0.002	0.005	0.696	0.003	0.006	0.882	0.003	0.005
tf-idf	0.741	0.003	0.007	0.464	0.002	0.004	0.667	0.012	0.023

Table 4: Score-thresholding results of untagged word patterns

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Threshold score > 0.01									
IG	0.444	0.328	0.377	0.084	0.072	0.078	0.145	0.119	0.131
IGR	0.439	0.377	0.406	0.084	0.072	0.078	0.142	0.126	0.133
MI	0.436	0.684	0.533	0.032	0.173	0.054	0.103	0.700	0.179
NMI	0.439	0.593	0.504	0.035	0.164	0.058	0.106	0.620	0.182
LL	0.439	0.378	0.407	0.084	0.072	0.078	0.142	0.123	0.131
CHI	0.439	0.266	0.331	0.221	0.056	0.090	0.349	0.065	0.110
Meta	0.436	0.398	0.416	0.036	0.123	0.056	0.108	0.467	0.176
tf-idf	0.414	0.311	0.355	0.042	0.140	0.064	0.110	0.439	0.176
Threshold score > 0.02									
IG	0.457	0.239	0.314	0.166	0.043	0.068	0.203	0.066	0.100
IGR	0.457	0.242	0.316	0.156	0.044	0.069	0.203	0.066	0.100
MI	0.438	0.582	0.500	0.034	0.157	0.056	0.106	0.610	0.180
NMI	0.436	0.513	0.471	0.037	0.146	0.059	0.109	0.551	0.182
LL	0.457	0.242	0.316	0.156	0.044	0.069	0.203	0.066	0.099
CHI	0.442	0.213	0.287	0.364	0.035	0.064	0.467	0.045	0.083

Meta	0.438	0.349	0.389	0.041	0.104	0.059	0.115	0.372	0.175
tf-idf	0.427	0.278	0.337	0.047	0.115	0.067	0.112	0.374	0.173
Threshold score > 0.03									
IG	0.729	0.056	0.104	0.209	0.032	0.056	0.258	0.046	0.078
IGR	0.731	0.057	0.107	0.209	0.032	0.056	0.252	0.044	0.074
MI	0.443	0.526	0.481	0.035	0.141	0.056	0.107	0.557	0.180
NMI	0.435	0.433	0.434	0.039	0.130	0.060	0.112	0.470	0.181
LL	0.730	0.057	0.106	0.209	0.032	0.056	0.253	0.044	0.075
CHI	0.737	0.038	0.072	0.412	0.026	0.049	0.497	0.031	0.058
Meta	0.440	0.301	0.357	0.046	0.093	0.061	0.121	0.316	0.175
tf-idf	0.418	0.183	0.255	0.053	0.103	0.070	0.123	0.349	0.182
Threshold score > 0.04									
IG	0.749	0.047	0.088	0.213	0.026	0.046	0.276	0.035	0.062
IGR	0.749	0.047	0.088	0.215	0.026	0.047	0.276	0.035	0.062
MI	0.434	0.498	0.463	0.035	0.137	0.056	0.108	0.540	0.180
NMI	0.433	0.395	0.413	0.039	0.121	0.059	0.112	0.442	0.179
LL	0.749	0.047	0.088	0.215	0.026	0.047	0.276	0.035	0.062
CHI	0.751	0.033	0.063	0.559	0.020	0.039	0.517	0.024	0.047
Meta	0.439	0.267	0.332	0.058	0.081	0.068	0.139	0.253	0.179
tf-idf	0.465	0.165	0.244	0.062	0.077	0.069	0.124	0.328	0.180
Threshold score > 0.05									
IG	0.816	0.033	0.063	0.247	0.021	0.040	0.318	0.029	0.053
IGR	0.817	0.031	0.060	0.246	0.021	0.039	0.318	0.029	0.053
MI	0.434	0.444	0.439	0.036	0.126	0.056	0.109	0.487	0.177
NMI	0.436	0.374	0.402	0.039	0.116	0.058	0.112	0.430	0.178
LL	0.817	0.031	0.060	0.246	0.021	0.039	0.318	0.029	0.053
CHI	0.743	0.031	0.060	0.655	0.017	0.033	0.709	0.019	0.037
Meta	0.447	0.219	0.294	0.056	0.075	0.064	0.138	0.245	0.177
tf-idf	0.499	0.150	0.231	0.068	0.064	0.066	0.124	0.317	0.179
Threshold score > 0.06									
IG	0.809	0.028	0.053	0.319	0.018	0.034	0.310	0.025	0.046
IGR	0.827	0.026	0.051	0.319	0.018	0.034	0.310	0.025	0.046
MI	0.430	0.422	0.426	0.037	0.120	0.056	0.110	0.458	0.177
NMI	0.432	0.337	0.379	0.041	0.109	0.060	0.114	0.394	0.177
LL	0.827	0.026	0.051	0.319	0.018	0.034	0.310	0.025	0.046

CHI	0.878	0.018	0.035	0.683	0.015	0.029	0.750	0.016	0.031
Meta	0.444	0.216	0.290	0.177	0.057	0.087	0.236	0.094	0.135
tf-idf	0.500	0.094	0.158	0.108	0.042	0.060	0.151	0.268	0.194
Threshold score > 0.07									
IG	0.852	0.024	0.047	0.307	0.016	0.030	0.361	0.021	0.041
IGR	0.876	0.020	0.039	0.308	0.016	0.030	0.360	0.021	0.040
MI	0.431	0.410	0.420	0.036	0.115	0.055	0.109	0.444	0.176
NMI	0.431	0.334	0.376	0.040	0.105	0.058	0.114	0.387	0.176
LL	0.876	0.020	0.039	0.308	0.016	0.030	0.360	0.021	0.040
CHI	0.873	0.017	0.033	0.719	0.012	0.024	0.796	0.014	0.028
Meta	0.441	0.212	0.286	0.178	0.055	0.084	0.239	0.090	0.131
tf-idf	0.542	0.071	0.125	0.119	0.033	0.052	0.200	0.246	0.221
Threshold score > 0.08									
IG	0.718	0.018	0.035	0.296	0.014	0.027	0.354	0.020	0.037
IGR	0.720	0.019	0.037	0.297	0.014	0.027	0.351	0.020	0.037
MI	0.430	0.388	0.408	0.036	0.111	0.054	0.110	0.432	0.175
NMI	0.431	0.329	0.373	0.040	0.102	0.058	0.114	0.375	0.175
LL	0.720	0.019	0.037	0.297	0.014	0.027	0.354	0.020	0.037
CHI	**0.921**	0.012	0.024	**0.742**	0.012	0.023	**0.835**	0.013	0.025
Meta	0.724	0.041	0.078	0.201	0.050	0.079	0.267	0.081	0.124
tf-idf	0.601	0.069	0.124	0.141	0.024	0.041	0.257	0.182	0.213
Threshold score > 0.09									
IG	0.698	0.016	0.030	0.378	0.012	0.023	0.415	0.015	0.028
IGR	0.715	0.017	0.034	0.386	0.013	0.024	0.415	0.015	0.028
MI	0.427	0.379	0.402	0.036	0.106	0.053	0.109	0.419	0.173
NMI	0.433	0.307	0.360	0.041	0.099	0.058	0.115	0.362	0.175
LL	0.715	0.017	0.034	0.386	0.013	0.024	0.412	0.014	0.028
CHI	**0.955**	0.011	0.022	**0.773**	0.010	0.020	**0.829**	0.010	0.020
Meta	0.725	0.038	0.072	0.205	0.048	0.078	0.268	0.077	0.119
tf-idf	0.680	0.050	0.093	0.184	0.020	0.035	0.313	0.155	0.208
Threshold score > 0.1									
IG	0.692	0.015	0.029	0.359	0.011	0.021	0.402	0.014	0.027
IGR	0.697	0.015	0.029	0.359	0.011	0.021	0.402	0.014	0.027
MI	0.427	0.372	0.398	0.035	0.104	0.053	0.109	0.408	0.173
NMI	0.431	0.304	0.357	0.040	0.096	0.056	0.115	0.358	0.174

LL	0.697	0.015	0.029	0.359	0.011	0.021	0.402	0.014	0.027
CHI	**0.951**	0.010	0.020	**0.774**	0.008	0.017	**0.839**	0.009	0.018
Meta	0.739	0.034	0.066	0.272	0.040	0.070	0.349	0.055	0.095
tf-idf	0.700	0.036	0.068	0.276	0.015	0.028	0.376	0.119	0.180
Threshold score > 0.2									
IG	0.853	0.005	0.010	0.320	0.006	0.011	0.348	0.007	0.014
IGR	0.853	0.007	0.014	0.320	0.006	0.011	0.348	0.007	0.014
MI	0.427	0.297	0.350	0.035	0.082	0.049	0.111	0.339	0.167
NMI	0.439	0.209	0.283	0.054	0.072	0.062	0.136	0.239	0.173
LL	0.853	0.005	0.010	0.320	0.006	0.011	0.348	0.007	0.014
CHI	**1.000**	0.004	0.007	**0.806**	0.004	0.009	**0.839**	0.005	0.009
Meta	0.894	0.010	0.021	0.400	0.014	0.027	0.495	0.019	0.037
tf-idf	0.775	0.025	0.048	0.300	0.010	0.019	0.535	0.086	0.148
Threshold score > 0.3									
IG	0.810	0.003	0.006	0.250	0.004	0.008	0.293	0.005	0.010
IGR	0.810	0.003	0.006	0.250	0.004	0.008	0.293	0.005	0.010
MI	0.424	0.275	0.334	0.035	0.074	0.048	0.112	0.305	0.164
NMI	0.729	0.034	0.064	0.173	0.050	0.078	0.236	0.085	0.125
LL	0.810	0.003	0.006	0.250	0.004	0.008	0.293	0.005	0.010
CHI	**1.000**	0.002	0.004	**0.850**	0.003	0.006	**0.938**	0.003	0.005
Meta	1.000	0.004	0.008	0.833	0.006	0.012	0.837	0.006	0.013
tf-idf	0.854	0.014	0.028	0.385	0.007	0.015	0.695	0.052	0.096
Threshold score > 0.4									
IG	1.000	0.002	0.004	0.203	0.003	0.006	0.239	0.004	0.008
IGR	1.000	0.002	0.004	0.203	0.003	0.006	0.239	0.004	0.008
MI	0.422	0.245	0.310	0.036	0.068	0.047	0.114	0.277	0.161
NMI	0.722	0.026	0.050	0.199	0.042	0.070	0.266	0.071	0.112
LL	1.000	0.002	0.004	0.203	0.003	0.006	0.239	0.004	0.008
CHI	**1.000**	0.002	0.003	0.909	0.002	0.004	0.917	0.002	0.004
Meta	1.000	0.002	0.004	0.826	0.003	0.007	0.895	0.003	0.006
tf-idf	0.906	0.005	0.010	0.453	0.004	0.008	0.830	0.016	0.032

Table 5: Score-thresholding results of PoS-tagged word patterns

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Threshold score > 0.01									
IG	0.447	0.361	0.400	0.071	0.071	0.071	0.147	0.200	0.169
IGR	0.444	0.455	0.449	0.069	0.072	0.070	0.152	0.179	0.164
MI	0.451	0.689	0.545	0.030	0.156	0.050	0.108	0.722	0.188
NMI	0.448	0.589	0.509	0.031	0.147	0.052	0.110	0.650	0.189
LL	0.444	0.455	0.449	0.068	0.073	0.071	0.152	0.179	0.164
CHI	0.444	0.291	0.352	0.099	0.055	0.071	0.203	0.119	0.150
Meta	0.448	0.511	0.478	0.032	0.111	0.050	0.113	0.494	0.184
tf-idf	0.415	0.504	0.455	0.036	0.108	0.054	0.115	0.458	0.183
Threshold score > 0.02									
IG	0.460	0.233	0.309	0.125	0.050	0.072	0.177	0.086	0.116
IGR	0.450	0.307	0.365	0.124	0.050	0.071	0.180	0.084	0.115
MI	0.446	0.572	0.501	0.031	0.140	0.051	0.110	0.632	0.187
NMI	0.448	0.507	0.476	0.033	0.131	0.052	0.113	0.578	0.189
LL	0.450	0.307	0.365	0.125	0.050	0.072	0.187	0.083	0.115
CHI	0.445	0.213	0.288	0.164	0.039	0.063	0.277	0.054	0.091
Meta	0.448	0.357	0.397	0.036	0.094	0.052	0.117	0.393	0.180
tf-idf	0.435	0.504	0.467	0.042	0.108	0.061	0.118	0.423	0.184
Threshold score > 0.03									
IG	0.776	0.054	0.100	0.176	0.038	0.062	0.268	0.051	0.086
IGR	0.457	0.228	0.305	0.183	0.037	0.062	0.270	0.046	0.079
MI	0.446	0.513	0.477	0.031	0.126	0.050	0.111	0.579	0.186
NMI	0.443	0.430	0.436	0.033	0.116	0.052	0.115	0.506	0.187
LL	0.457	0.228	0.305	0.179	0.038	0.062	0.270	0.046	0.079
CHI	0.440	0.209	0.283	0.211	0.030	0.053	0.331	0.042	0.075
Meta	0.448	0.315	0.370	0.040	0.085	0.054	0.121	0.334	0.178
tf-idf	0.435	0.383	0.407	0.045	0.076	0.056	0.126	0.352	0.186
Threshold score > 0.04									
IG	0.770	0.047	0.088	0.230	0.028	0.049	0.313	0.036	0.064
IGR	0.775	0.051	0.095	0.232	0.028	0.050	0.312	0.035	0.063
MI	0.444	0.493	0.467	0.031	0.122	0.050	0.112	0.553	0.186
NMI	0.444	0.390	0.416	0.034	0.108	0.051	0.115	0.470	0.184
LL	0.775	0.051	0.095	0.224	0.028	0.050	0.312	0.035	0.063

CHI	0.760	0.034	0.065	0.289	0.025	0.045	0.356	0.028	0.052
Meta	0.445	0.293	0.353	0.048	0.074	0.058	0.136	0.275	0.182
tf-idf	0.594	0.226	0.327	0.050	0.067	0.057	0.132	0.274	0.178
Threshold score > 0.05									
IG	0.770	0.041	0.078	0.245	0.023	0.042	0.350	0.029	0.053
IGR	0.768	0.045	0.084	0.245	0.023	0.042	0.350	0.029	0.053
MI	0.441	0.441	0.441	0.031	0.111	0.048	0.112	0.509	0.184
NMI	0.445	0.368	0.403	0.033	0.102	0.050	0.115	0.455	0.184
LL	0.768	0.045	0.084	0.245	0.023	0.042	0.350	0.029	0.053
CHI	0.764	0.030	0.058	0.360	0.019	0.036	0.488	0.021	0.040
Meta	0.445	0.265	0.332	0.047	0.070	0.056	0.136	0.266	0.180
tf-idf	0.627	0.187	0.288	0.055	0.055	0.055	0.141	0.240	0.178
Threshold score > 0.06									
IG	0.762	0.038	0.073	0.263	0.020	0.038	0.347	0.024	0.044
IGR	0.766	0.039	0.074	0.263	0.020	0.038	0.347	0.024	0.044
MI	0.439	0.421	0.429	0.031	0.105	0.048	0.113	0.481	0.182
NMI	0.441	0.334	0.380	0.034	0.097	0.050	0.116	0.423	0.182
LL	0.766	0.039	0.074	0.263	0.020	0.038	0.347	0.024	0.044
CHI	0.758	0.028	0.054	0.404	0.017	0.032	0.565	0.018	0.036
Meta	0.446	0.216	0.291	0.090	0.053	0.066	0.183	0.131	0.153
tf-idf	0.658	0.130	0.217	0.068	0.048	0.057	0.192	0.162	0.176
Threshold score > 0.07									
IG	0.864	0.025	0.049	0.336	0.018	0.034	0.409	0.021	0.040
IGR	0.856	0.027	0.052	0.340	0.017	0.033	0.409	0.021	0.040
MI	0.439	0.406	0.422	0.031	0.102	0.048	0.112	0.468	0.181
NMI	0.441	0.332	0.379	0.034	0.094	0.050	0.116	0.408	0.180
LL	0.856	0.027	0.052	0.340	0.017	0.033	0.409	0.021	0.040
CHI	0.889	0.016	0.031	0.455	0.015	0.029	0.610	0.016	0.031
Meta	0.445	0.214	0.289	0.095	0.051	0.067	0.183	0.125	0.149
tf-idf	0.686	0.076	0.137	0.079	0.036	0.050	0.202	0.136	0.163
Threshold score > 0.08									
IG	0.840	0.021	0.041	0.348	0.016	0.030	0.410	0.019	0.036
IGR	0.850	0.017	0.032	0.350	0.016	0.031	0.410	0.019	0.036
MI	0.443	0.386	0.412	0.031	0.098	0.047	0.113	0.453	0.180
NMI	0.440	0.327	0.375	0.034	0.091	0.049	0.116	0.394	0.179

LL	0.850	0.023	0.044	0.352	0.016	0.031	0.410	0.019	0.036
CHI	**0.881**	0.015	0.029	**0.524**	0.013	0.026	**0.630**	0.014	0.028
Meta	0.762	0.040	0.076	0.110	0.047	0.065	0.198	0.108	0.140
tf-idf	0.725	0.026	0.050	0.101	0.024	0.039	0.233	0.107	0.146
Threshold score > 0.09									
IG	0.859	0.020	0.039	0.345	0.015	0.028	0.412	0.017	0.033
IGR	0.844	0.020	0.039	0.345	0.015	0.028	0.415	0.018	0.034
MI	0.439	0.378	0.406	0.031	0.094	0.046	0.111	0.435	0.177
NMI	0.442	0.305	0.361	0.035	0.089	0.050	0.117	0.383	0.179
LL	0.844	0.020	0.039	0.345	0.015	0.028	0.415	0.018	0.034
CHI	**0.875**	0.014	0.027	**0.578**	0.012	0.023	**0.689**	0.013	0.025
Meta	0.758	0.039	0.074	0.111	0.045	0.064	0.201	0.105	0.138
tf-idf	0.753	0.014	0.028	0.134	0.020	0.035	0.240	0.090	0.131
Threshold score > 0.1									
IG	0.889	0.014	0.027	0.353	0.013	0.025	0.404	0.016	0.031
IGR	0.891	0.014	0.028	0.345	0.013	0.025	0.411	0.017	0.032
MI	0.439	0.371	0.402	0.031	0.091	0.046	0.112	0.429	0.177
NMI	0.440	0.302	0.358	0.034	0.085	0.048	0.116	0.372	0.176
LL	0.891	0.014	0.028	0.345	0.013	0.025	0.411	0.017	0.032
CHI	**0.947**	0.009	0.018	**0.608**	0.011	0.022	**0.758**	0.012	0.023
Meta	0.754	0.036	0.069	0.109	0.042	0.061	0.199	0.100	0.133
tf-idf	0.783	0.009	0.018	0.215	0.017	0.031	0.268	0.073	0.115
Threshold score > 0.2									
IG	0.852	0.006	0.011	0.341	0.007	0.014	0.389	0.009	0.018
IGR	0.852	0.006	0.011	0.341	0.007	0.014	0.389	0.009	0.018
MI	0.434	0.318	0.367	0.029	0.074	0.042	0.111	0.358	0.170
NMI	0.440	0.208	0.283	0.044	0.064	0.052	0.134	0.260	0.176
LL	0.852	0.006	0.011	0.341	0.007	0.014	0.389	0.009	0.018
CHI	**1.000**	0.003	0.006	**0.800**	0.006	0.012	**0.852**	0.006	0.011
Meta	0.867	0.010	0.019	0.310	0.015	0.029	0.396	0.019	0.037
tf-idf	0.826	0.005	0.009	0.393	0.012	0.023	0.411	0.044	0.079
Threshold score > 0.3									
IG	0.789	0.004	0.007	0.280	0.005	0.010	0.326	0.007	0.014
IGR	0.789	0.004	0.007	0.280	0.005	0.010	0.326	0.007	0.014
MI	0.432	0.275	0.336	0.032	0.068	0.043	0.114	0.311	0.166

NMI	0.754	0.031	0.060	0.091	0.046	0.061	0.178	0.117	0.141
LL	0.789	0.004	0.007	0.280	0.005	0.010	0.326	0.007	0.014
CHI	**1.000**	0.002	0.004	**0.789**	0.004	0.007	**0.833**	0.004	0.007
Meta	1.000	0.004	0.008	0.727	0.006	0.012	0.405	0.008	0.016
tf-idf	0.846	0.003	0.005	0.500	0.008	0.016	0.572	0.032	0.060
Threshold score > 0.4									
IG	0.714	0.002	0.005	0.254	0.004	0.009	0.316	0.006	0.012
IGR	0.714	0.002	0.004	0.254	0.004	0.009	0.316	0.006	0.012
MI	0.431	0.247	0.314	0.033	0.062	0.043	0.116	0.283	0.164
NMI	0.735	0.025	0.048	0.100	0.037	0.054	0.195	0.095	0.128
LL	0.714	0.002	0.005	0.254	0.004	0.009	0.316	0.006	0.012
CHI	**1.000**	0.001	0.003	**0.867**	0.003	0.006	1.000	0.002	0.005
Meta	1.000	0.002	0.005	0.810	0.004	0.008	0.842	0.004	0.008
tf-idf	0.875	0.002	0.003	0.677	0.005	0.010	0.764	0.010	0.021

Table 6: Score-thresholding results of verb-centred word patterns

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Top 100 Ranked Patterns									
IG	0.720	0.015	0.029	0.030	0.001	0.001	0.030	0.001	0.001
IGR	0.740	0.015	0.030	0.030	0.001	0.001	0.030	0.001	0.001
MI	0.400	0.008	0.016	0.010	0.000	0.000	0.180	0.004	0.007
NMI	0.770	0.016	0.031	0.190	0.004	0.008	**0.250**	0.005	0.010
LL	0.740	0.015	0.030	0.030	0.001	0.001	0.030	0.001	0.001
CHI	0.770	0.016	0.031	**0.220**	0.005	0.009	0.200	0.004	0.008
Meta	**0.770**	0.016	0.031	0.120	0.002	0.005	0.100	0.002	0.004
tf-idf	0.750	0.015	0.030	0.130	0.003	0.005	0.170	0.004	0.007
Top 200 Ranked Patterns									
IG	0.670	0.028	0.053	0.025	0.001	0.002	0.050	0.002	0.004
IGR	0.680	0.028	0.054	0.025	0.001	0.002	0.050	0.002	0.004
MI	0.380	0.016	0.030	0.020	0.000	0.001	0.135	0.006	0.011
NMI	0.575	0.024	0.046	0.130	0.005	0.010	0.195	0.008	0.015
LL	0.680	0.028	0.054	0.025	0.001	0.002	0.050	0.002	0.004
CHI	0.630	0.026	0.050	**0.150**	0.006	0.012	**0.200**	0.008	0.016
Meta	**0.705**	0.029	0.056	0.085	0.004	0.007	0.090	0.004	0.007

tf-idf	0.670	0.028	0.053	0.100	0.004	0.008	0.140	0.006	0.011
Top 300 Ranked Patterns									
IG	0.607	0.037	0.071	0.047	0.003	0.005	0.067	0.004	0.008
IGR	0.627	0.039	0.073	0.047	0.003	0.005	0.070	0.004	0.008
MI	0.443	0.027	0.052	0.010	0.001	0.001	0.140	0.009	0.016
NMI	0.500	0.031	0.058	**0.147**	0.009	0.017	**0.180**	0.011	0.021
LL	**0.657**	0.041	0.076	0.047	0.003	0.005	0.070	0.004	0.008
CHI	0.537	0.033	0.062	0.133	0.008	0.016	0.173	0.011	0.020
Meta	0.640	0.040	0.075	0.083	0.005	0.010	0.107	0.007	0.012
tf-idf	0.613	0.038	0.071	0.097	0.006	0.011	0.110	0.007	0.013

Table 7: Rank-thresholding results of untagged word patterns along with prepositions

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Top 100 Ranked Patterns									
IG	0.610	0.014	0.027	0.100	0.002	0.004	0.110	0.002	0.005
IGR	0.610	0.014	0.027	0.110	0.002	0.005	0.110	0.002	0.005
MI	0.440	0.010	0.019	0.010	0.000	0.000	0.050	0.001	0.002
NMI	0.690	0.016	0.030	0.320	0.007	0.014	0.350	0.008	0.015
LL	0.610	0.014	0.027	0.110	0.002	0.005	0.110	0.002	0.005
CHI	0.740	0.017	0.033	**0.380**	0.009	0.017	**0.430**	0.010	0.019
Meta	**0.760**	0.017	0.033	0.240	0.005	0.011	0.240	0.005	0.011
tf-idf	0.670	0.015	0.029	0.260	0.006	0.011	0.290	0.007	0.013
Top 200 Ranked Patterns									
IG	**0.655**	0.029	0.056	0.135	0.006	0.012	0.175	0.008	0.015
IGR	0.650	0.029	0.056	0.130	0.006	0.011	0.170	0.008	0.015
MI	0.465	0.021	0.040	0.020	0.001	0.002	0.060	0.003	0.005
NMI	0.635	0.029	0.055	0.205	0.009	0.018	**0.265**	0.012	0.023
LL	0.650	0.029	0.056	0.130	0.006	0.011	0.170	0.008	0.015
CHI	0.655	0.029	0.056	**0.220**	0.010	0.019	0.235	0.011	0.020
Meta	0.650	0.029	0.056	0.185	0.008	0.016	0.200	0.009	0.017
tf-idf	0.650	0.029	0.056	0.185	0.008	0.016	0.215	0.010	0.019
Top 300 Ranked Patterns									
IG	**0.667**	0.045	0.084	0.103	0.007	0.013	0.137	0.009	0.017

IGR	0.643	0.043	0.081	0.103	0.007	0.013	0.137	0.009	0.017
MI	0.470	0.032	0.059	0.023	0.002	0.003	0.063	0.004	0.008
NMI	0.550	0.037	0.070	0.173	0.012	0.022	**0.233**	0.016	0.029
LL	0.643	0.043	0.081	0.103	0.007	0.013	0.133	0.009	0.017
CHI	0.563	0.038	0.071	**0.177**	0.012	0.022	0.227	0.015	0.029
Meta	0.607	0.041	0.077	0.157	0.011	0.020	0.200	0.013	0.025
tf-idf	0.603	0.041	0.076	0.143	0.010	0.018	0.177	0.012	0.022

Table 8: Rank-thresholding results of PoS-tagged word patterns along with prepositions

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Top 100 Ranked Patterns									
IG	0.620	0.019	0.037	0.180	0.006	0.011	0.140	0.004	0.008
IGR	0.620	0.019	0.037	0.180	0.006	0.011	0.140	0.004	0.008
MI	0.430	0.013	0.026	0.030	0.001	0.002	0.030	0.001	0.002
NMI	0.690	0.021	0.041	0.300	0.009	0.018	**0.360**	0.011	0.021
LL	0.620	0.019	0.037	0.180	0.006	0.011	0.140	0.004	0.008
CHI	0.700	0.021	0.042	**0.350**	0.011	0.021	0.330	0.010	0.020
Meta	**0.760**	0.023	0.045	0.230	0.007	0.014	0.200	0.006	0.012
tf-idf	0.660	0.020	0.039	0.240	0.007	0.014	0.260	0.008	0.015
Top 200 Ranked Patterns									
IG	**0.690**	0.042	0.080	0.130	0.008	0.015	0.150	0.001	0.002
IGR	0.670	0.041	0.077	0.140	0.009	0.016	0.155	0.001	0.002
MI	0.445	0.027	0.051	0.015	0.001	0.002	0.050	0.003	0.006
NMI	0.545	0.033	0.063	0.215	0.013	0.025	**0.285**	0.017	0.033
LL	0.670	0.041	0.077	0.140	0.009	0.016	0.155	0.001	0.002
CHI	0.560	0.034	0.065	**0.225**	0.014	0.026	0.255	0.016	0.029
Meta	0.610	0.037	0.070	0.200	0.012	0.023	0.225	0.014	0.026
tf-idf	0.605	0.037	0.070	0.210	0.013	0.024	0.220	0.013	0.025
Top 300 Ranked Patterns									
IG	**0.607**	0.056	0.102	0.093	0.009	0.016	0.103	0.009	0.017
IGR	0.600	0.055	0.101	0.093	0.009	0.016	0.107	0.010	0.018
MI	0.470	0.043	0.079	0.020	0.002	0.003	0.050	0.005	0.008
NMI	0.503	0.064	0.113	0.173	0.016	0.029	**0.233**	0.021	0.039

LL	0.600	0.055	0.101	0.093	0.009	0.016	0.110	0.010	0.018
CHI	0.533	0.049	0.090	**0.190**	0.017	0.032	0.230	0.021	0.039
Meta	0.543	0.050	0.091	0.167	0.015	0.028	0.213	0.020	0.036
tf-idf	0.533	0.049	0.090	0.170	0.016	0.029	0.187	0.017	0.031

Table 9: Rank-thresholding results of verb-centred word patterns along with prepositions

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Threshold score > 0.01									
IG	0.449	0.693	0.545	0.018	0.043	0.025	0.075	0.230	0.113
IGR	0.448	0.759	0.563	0.018	0.044	0.026	0.075	0.227	0.112
MI	0.444	0.776	0.564	0.012	0.068	0.020	0.068	0.484	0.119
NMI	0.449	0.728	0.555	0.012	0.064	0.020	0.068	0.450	0.119
LL	0.448	0.759	0.563	0.018	0.043	0.025	0.075	0.226	0.112
CHI	0.466	0.488	0.477	0.020	0.033	0.025	0.076	0.162	0.103
Meta	0.450	0.728	0.556	0.012	0.058	0.020	0.067	0.420	0.115
tf-idf	0.435	0.684	0.532	0.009	0.051	0.015	0.061	0.410	0.107
Threshold score > 0.02									
IG	0.449	0.693	0.545	0.027	0.032	0.029	0.078	0.100	0.088
IGR	0.450	0.716	0.552	0.027	0.032	0.029	0.079	0.102	0.089
MI	0.452	0.700	0.549	0.012	0.061	0.020	0.068	0.445	0.118
NMI	0.454	0.657	0.537	0.012	0.058	0.020	0.068	0.408	0.117
LL	0.450	0.716	0.552	0.027	0.032	0.030	0.079	0.102	0.089
CHI	0.470	0.405	0.435	0.045	0.024	0.031	0.089	0.048	0.062
Meta	0.449	0.692	0.545	0.012	0.048	0.019	0.066	0.319	0.109
tf-idf	0.433	0.650	0.520	0.011	0.048	0.018	0.063	0.377	0.108
Threshold score > 0.03									
IG	0.449	0.693	0.545	0.031	0.025	0.028	0.081	0.071	0.076
IGR	0.457	0.577	0.510	0.030	0.022	0.026	0.081	0.071	0.076
MI	0.453	0.653	0.535	0.012	0.058	0.019	0.068	0.421	0.117
NMI	0.463	0.522	0.491	0.012	0.054	0.020	0.068	0.369	0.114
LL	0.457	0.577	0.510	0.031	0.025	0.028	0.081	0.071	0.076
CHI	0.468	0.345	0.398	0.062	0.020	0.030	0.110	0.036	0.054
Meta	0.459	0.536	0.495	0.013	0.043	0.019	0.069	0.249	0.107

tf-idf	0.414	0.487	0.448	0.012	0.045	0.019	0.066	0.329	0.110
Threshold score > 0.04									
IG	0.449	0.693	0.545	0.030	0.020	0.024	0.078	0.057	0.066
IGR	0.462	0.520	0.490	0.031	0.020	0.024	0.077	0.056	0.065
MI	0.460	0.537	0.496	0.012	0.056	0.019	0.068	0.400	0.116
NMI	0.464	0.498	0.481	0.013	0.052	0.020	0.067	0.349	0.113
LL	0.462	0.522	0.490	0.031	0.020	0.024	0.078	0.057	0.066
CHI	0.467	0.295	0.362	0.077	0.016	0.026	0.118	0.033	0.052
Meta	0.461	0.500	0.480	0.014	0.039	0.021	0.069	0.235	0.106
tf-idf	0.430	0.402	0.416	0.012	0.041	0.019	0.068	0.291	0.111
Threshold score > 0.05									
IG	0.466	0.372	0.414	0.031	0.018	0.023	0.079	0.049	0.061
IGR	0.463	0.490	0.476	0.031	0.017	0.022	0.079	0.049	0.061
MI	0.465	0.512	0.487	0.012	0.053	0.019	0.068	0.387	0.115
NMI	0.466	0.482	0.474	0.012	0.049	0.020	0.067	0.335	0.111
LL	0.463	0.491	0.477	0.030	0.017	0.021	0.079	0.049	0.061
CHI	0.468	0.294	0.361	0.099	0.015	0.026	0.128	0.017	0.030
Meta	0.467	0.440	0.453	0.020	0.032	0.024	0.076	0.157	0.103
tf-idf	0.442	0.314	0.367	0.013	0.034	0.018	0.074	0.223	0.111
Threshold score > 0.06									
IG	0.466	0.372	0.414	0.032	0.014	0.019	0.082	0.039	0.053
IGR	0.468	0.428	0.447	0.035	0.016	0.022	0.079	0.040	0.053
MI	0.464	0.496	0.480	0.012	0.051	0.019	0.067	0.365	0.114
NMI	0.464	0.463	0.463	0.012	0.046	0.019	0.065	0.308	0.108
LL	0.468	0.428	0.447	0.035	0.016	0.022	0.079	0.040	0.053
CHI	0.467	0.269	0.341	0.111	0.012	0.022	0.139	0.014	0.025
Meta	0.468	0.381	0.420	0.020	0.030	0.024	0.077	0.149	0.101
tf-idf	0.445	0.256	0.325	0.014	0.027	0.019	0.076	0.178	0.106
Threshold score > 0.07									
IG	0.466	0.372	0.414	0.030	0.012	0.017	0.082	0.037	0.051
IGR	0.466	0.372	0.414	0.030	0.012	0.017	0.081	0.036	0.050
MI	0.461	0.486	0.473	0.012	0.050	0.019	0.067	0.356	0.113
NMI	0.468	0.414	0.439	0.012	0.044	0.019	0.064	0.297	0.106
LL	0.466	0.372	0.414	0.030	0.012	0.017	0.081	0.036	0.050
CHI	0.464	0.263	0.336	0.121	0.011	0.020	0.141	0.012	0.022

Meta	0.464	0.352	0.400	0.041	0.024	0.030	0.087	0.056	0.068
tf-idf	0.453	0.230	0.305	0.016	0.024	0.019	0.084	0.123	0.100
Threshold score > 0.08									
IG	0.466	0.372	0.414	0.030	0.011	0.016	0.087	0.034	0.049
IGR	0.464	0.318	0.377	0.030	0.011	0.016	0.087	0.034	0.049
MI	0.462	0.472	0.467	0.012	0.049	0.019	0.067	0.342	0.112
NMI	**0.470**	0.401	0.433	0.012	0.044	0.019	0.065	0.285	0.106
LL	0.464	0.318	0.377	0.030	0.011	0.016	0.087	0.034	0.049
CHI	0.464	0.262	0.335	**0.121**	0.008	0.016	**0.141**	0.011	0.021
Meta	0.467	0.332	0.388	0.043	0.023	0.030	0.088	0.054	0.067
tf-idf	0.461	0.201	0.280	0.018	0.020	0.019	0.085	0.098	0.091
Threshold score > 0.09									
IG	0.466	0.372	0.414	0.033	0.011	0.016	0.085	0.030	0.044
IGR	0.466	0.283	0.352	0.033	0.011	0.016	0.086	0.031	0.045
MI	0.462	0.461	0.462	0.012	0.047	0.019	0.067	0.340	0.112
NMI	0.467	0.363	0.408	0.013	0.042	0.020	0.066	0.268	0.106
LL	0.466	0.283	0.352	0.033	0.011	0.016	0.086	0.031	0.045
CHI	**0.470**	0.172	0.252	**0.143**	0.007	0.014	**0.187**	0.009	0.018
Meta	0.467	0.301	0.366	0.048	0.020	0.029	0.074	0.045	0.056
tf-idf	0.466	0.191	0.271	0.020	0.018	0.019	0.090	0.052	0.066
Threshold score > 0.1									
IG	0.469	0.189	0.269	0.031	0.009	0.014	0.085	0.027	0.041
IGR	0.464	0.280	0.349	0.031	0.009	0.014	0.085	0.027	0.041
MI	0.464	0.454	0.459	0.012	0.046	0.019	0.066	0.328	0.110
NMI	0.465	0.341	0.393	0.012	0.041	0.019	0.066	0.267	0.105
LL	0.465	0.280	0.349	0.031	0.009	0.014	0.085	0.027	0.041
CHI	**0.470**	0.172	0.252	**0.140**	0.007	0.013	**0.198**	0.008	0.015
Meta	0.466	0.275	0.346	0.049	0.020	0.028	0.092	0.043	0.059
tf-idf	0.460	0.184	0.262	0.021	0.013	0.016	0.093	0.048	0.063
Threshold score > 0.2									
IG	0.670	0.028	0.054	0.032	0.004	0.007	0.100	0.017	0.029
IGR	0.670	0.028	0.054	0.032	0.004	0.007	0.100	0.017	0.029
MI	0.464	0.314	0.374	0.011	0.038	0.017	0.064	0.277	0.104
NMI	0.462	0.260	0.333	0.022	0.028	0.024	0.078	0.130	0.097
LL	0.670	0.028	0.054	0.032	0.004	0.007	0.100	0.017	0.029

CHI	**0.773**	0.018	0.034	**0.276**	0.003	0.007	**0.278**	0.003	0.006
Meta	0.735	0.023	0.046	0.083	0.007	0.014	0.115	0.010	0.018
tf-idf	0.539	0.074	0.129	0.021	0.011	0.014	0.111	0.036	0.055
Threshold score > 0.3									
IG	0.738	0.012	0.024	0.052	0.003	0.005	0.049	0.002	0.005
IGR	0.738	0.012	0.024	0.052	0.003	0.005	0.049	0.002	0.005
MI	0.460	0.280	0.348	0.012	0.034	0.018	0.065	0.227	0.101
NMI	0.463	0.166	0.244	0.067	0.020	0.031	0.100	0.035	0.052
LL	0.738	0.012	0.024	0.052	0.003	0.005	0.049	0.002	0.005
CHI	**0.778**	0.012	0.023	**0.314**	0.002	0.004	**0.417**	0.002	0.004
Meta	0.730	0.017	0.033	0.071	0.004	0.007	0.086	0.004	0.007
tf-idf	0.586	0.046	0.084	0.025	0.009	0.013	0.144	0.019	0.034
Threshold score > 0.4									
IG	0.702	0.008	0.016	0.025	0.001	0.002	0.033	0.001	0.002
IGR	0.696	0.008	0.016	0.025	0.001	0.002	0.033	0.001	0.002
MI	0.457	0.253	0.325	0.013	0.029	0.018	0.068	0.198	0.102
NMI	0.753	0.014	0.028	0.083	0.016	0.026	0.111	0.025	0.040
LL	0.696	0.008	0.016	0.025	0.001	0.002	0.033	0.001	0.002
CHI	0.850	0.004	0.007	**0.250**	0.001	0.002	0.333	0.001	0.002
Meta	**0.892**	0.007	0.013	0.120	0.002	0.004	0.128	0.002	0.004
tf-idf	0.655	0.019	0.037	0.082	0.007	0.012	0.156	0.009	0.017

Table 10: Score-thresholding results of untagged word patterns along with prepositions

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Threshold score > 0.01									
IG	0.439	0.615	0.512	0.017	0.047	0.025	0.045	0.107	0.063
IGR	0.440	0.583	0.501	0.017	0.047	0.025	0.044	0.099	0.061
MI	0.444	0.696	0.542	0.012	0.071	0.021	0.038	0.265	0.066
NMI	0.444	0.648	0.527	0.013	0.067	0.022	0.038	0.246	0.067
LL	0.440	0.583	0.501	0.017	0.047	0.025	0.044	0.099	0.061
CHI	0.447	0.342	0.387	0.023	0.037	0.028	0.068	0.055	0.061
Meta	0.440	0.610	0.511	0.013	0.056	0.021	0.039	0.212	0.066
tf-idf	0.388	0.559	0.458	0.014	0.059	0.023	0.046	0.223	0.077

Threshold score > 0.02									
IG	0.447	0.379	0.410	0.024	0.033	0.027	0.046	0.067	0.054
IGR	0.443	0.449	0.446	0.024	0.032	0.028	0.046	0.066	0.054
MI	0.442	0.623	0.517	0.012	0.063	0.021	0.038	0.242	0.066
NMI	0.443	0.538	0.486	0.014	0.061	0.023	0.040	0.213	0.068
LL	0.443	0.449	0.446	0.024	0.032	0.028	0.046	0.066	0.054
CHI	0.450	0.229	0.303	0.075	0.023	0.035	0.119	0.031	0.050
Meta	0.437	0.442	0.439	0.015	0.046	0.023	0.041	0.155	0.064
tf-idf	0.391	0.538	0.453	0.017	0.049	0.025	0.049	0.190	0.078
Threshold score > 0.03									
IG	0.447	0.379	0.410	0.027	0.022	0.024	0.045	0.046	0.045
IGR	0.450	0.339	0.386	0.026	0.022	0.024	0.045	0.046	0.045
MI	0.439	0.560	0.492	0.013	0.061	0.022	0.039	0.220	0.067
NMI	0.441	0.446	0.444	0.015	0.056	0.024	0.042	0.188	0.068
LL	0.450	0.339	0.387	0.026	0.022	0.024	0.045	0.046	0.045
CHI	0.452	0.200	0.277	0.130	0.018	0.032	0.166	0.025	0.043
Meta	0.448	0.362	0.401	0.018	0.041	0.025	0.045	0.132	0.067
tf-idf	0.399	0.456	0.425	0.019	0.041	0.026	0.053	0.153	0.078
Threshold score > 0.04									
IG	0.458	0.217	0.295	0.029	0.020	0.023	0.047	0.038	0.042
IGR	0.455	0.248	0.321	0.028	0.020	0.023	0.047	0.038	0.042
MI	0.441	0.531	0.482	0.014	0.057	0.022	0.040	0.207	0.067
NMI	0.438	0.429	0.433	0.016	0.051	0.024	0.043	0.175	0.069
LL	0.455	0.248	0.321	0.028	0.020	0.023	0.047	0.038	0.042
CHI	0.449	0.197	0.273	0.155	0.016	0.029	0.199	0.020	0.037
Meta	0.449	0.327	0.378	0.021	0.038	0.027	0.053	0.116	0.072
tf-idf	0.399	0.456	0.425	0.022	0.037	0.027	0.056	0.117	0.075
Threshold score > 0.05									
IG	0.458	0.217	0.295	0.034	0.017	0.022	0.051	0.029	0.037
IGR	0.455	0.214	0.291	0.032	0.017	0.022	0.049	0.031	0.038
MI	0.441	0.496	0.467	0.014	0.054	0.022	0.041	0.197	0.068
NMI	0.440	0.410	0.425	0.015	0.048	0.023	0.043	0.167	0.068
LL	0.455	0.214	0.291	0.032	0.017	0.022	0.049	0.031	0.038
CHI	0.445	0.193	0.270	0.160	0.013	0.023	0.218	0.018	0.032
Meta	0.444	0.285	0.347	0.024	0.036	0.029	0.056	0.106	0.073

tf-idf	0.425	0.402	0.413	0.024	0.029	0.026	0.060	0.092	0.073
Threshold score > 0.06									
IG	0.673	0.041	0.078	0.034	0.015	0.020	0.053	0.027	0.036
IGR	0.674	0.041	0.078	0.034	0.015	0.020	0.052	0.027	0.036
MI	0.438	0.438	0.438	0.014	0.051	0.022	0.041	0.182	0.067
NMI	0.443	0.396	0.418	0.016	0.047	0.024	0.044	0.159	0.068
LL	0.663	0.042	0.078	0.034	0.015	0.020	0.053	0.027	0.036
CHI	0.732	0.020	0.039	0.185	0.012	0.022	0.227	0.016	0.029
Meta	0.449	0.236	0.309	0.024	0.035	0.028	0.093	0.048	0.063
tf-idf	0.457	0.316	0.374	0.025	0.025	0.025	0.078	0.067	0.072
Threshold score > 0.07									
IG	0.673	0.041	0.078	0.042	0.013	0.020	0.051	0.023	0.032
IGR	0.661	0.038	0.071	0.043	0.013	0.020	0.051	0.023	0.032
MI	0.434	0.423	0.428	0.015	0.050	0.022	0.042	0.176	0.067
NMI	0.446	0.348	0.391	0.017	0.045	0.024	0.044	0.149	0.068
LL	0.661	0.038	0.071	0.043	0.013	0.020	0.051	0.023	0.032
CHI	0.725	0.020	0.038	0.311	0.009	0.018	0.236	0.014	0.026
Meta	0.452	0.208	0.285	0.066	0.025	0.036	0.096	0.045	0.061
tf-idf	0.463	0.218	0.297	0.045	0.020	0.027	0.080	0.049	0.061
Threshold score > 0.08									
IG	0.643	0.033	0.063	0.072	0.013	0.022	0.064	0.020	0.031
IGR	0.643	0.033	0.063	0.072	0.013	0.022	0.064	0.020	0.031
MI	0.433	0.419	0.426	0.015	0.048	0.022	0.042	0.170	0.067
NMI	0.443	0.336	0.382	0.018	0.044	0.025	0.045	0.139	0.068
LL	0.643	0.033	0.063	0.072	0.013	0.022	0.064	0.020	0.031
CHI	**0.752**	0.018	0.035	**0.339**	0.009	0.017	**0.232**	0.013	0.024
Meta	0.450	0.205	0.282	0.066	0.023	0.034	0.106	0.040	0.058
tf-idf	0.509	0.124	0.199	0.060	0.013	0.022	0.101	0.039	0.056
Threshold score > 0.09									
IG	0.653	0.029	0.055	0.066	0.011	0.019	0.091	0.017	0.028
IGR	0.640	0.030	0.058	0.066	0.011	0.019	0.092	0.017	0.029
MI	0.440	0.413	0.426	0.015	0.047	0.023	0.042	0.166	0.067
NMI	0.442	0.334	0.381	0.018	0.042	0.026	0.046	0.132	0.068
LL	0.640	0.030	0.058	0.066	0.011	0.019	0.092	0.017	0.029
CHI	**0.743**	0.017	0.033	**0.376**	0.009	0.017	**0.461**	0.009	0.018

Meta	0.448	0.203	0.279	0.069	0.021	0.032	0.108	0.038	0.056
tf-idf	0.534	0.070	0.124	0.071	0.011	0.018	0.114	0.028	0.045
Threshold score > 0.1									
IG	0.653	0.029	0.055	0.067	0.011	0.019	0.092	0.017	0.028
IGR	0.649	0.028	0.054	0.067	0.011	0.019	0.092	0.017	0.028
MI	0.439	0.407	0.423	0.014	0.045	0.022	0.042	0.165	0.067
NMI	0.444	0.312	0.367	0.018	0.041	0.025	0.046	0.131	0.068
LL	0.649	0.028	0.054	0.067	0.011	0.019	0.092	0.017	0.028
CHI	**0.737**	0.016	0.031	**0.427**	0.008	0.015	**0.488**	0.009	0.017
Meta	0.649	0.031	0.059	0.080	0.020	0.033	0.123	0.036	0.056
tf-idf	0.559	0.045	0.083	0.094	0.008	0.015	0.141	0.020	0.035
Threshold score > 0.2									
IG	**0.970**	0.007	0.014	0.149	0.006	0.011	0.171	0.008	0.015
IGR	0.970	0.007	0.014	0.149	0.006	0.011	0.171	0.008	0.015
MI	0.441	0.306	0.361	0.015	0.035	0.021	0.044	0.123	0.065
NMI	0.446	0.194	0.271	0.024	0.031	0.027	0.059	0.101	0.075
LL	0.970	0.007	0.014	0.149	0.006	0.011	0.171	0.008	0.015
CHI	0.952	0.004	0.009	**0.629**	0.005	0.010	**0.688**	0.005	0.010
Meta	0.870	0.009	0.018	0.179	0.010	0.019	0.207	0.012	0.023
tf-idf	0.577	0.025	0.048	0.109	0.005	0.009	0.202	0.014	0.026
Threshold score > 0.3									
IG	1.000	0.003	0.007	0.136	0.004	0.008	0.144	0.006	0.012
IGR	1.000	0.003	0.007	0.136	0.004	0.008	0.144	0.006	0.012
MI	0.436	0.264	0.329	0.016	0.032	0.021	0.044	0.113	0.064
NMI	0.703	0.018	0.034	0.076	0.019	0.030	0.120	0.036	0.055
LL	1.000	0.003	0.007	0.136	0.004	0.008	0.144	0.006	0.012
CHI	1.000	0.002	0.005	**0.652**	0.003	0.007	**0.696**	0.004	0.007
Meta	0.955	0.005	0.009	0.193	0.006	0.012	0.204	0.008	0.015
tf-idf	0.637	0.016	0.032	0.168	0.004	0.007	0.280	0.010	0.020
Threshold score > 0.4									
IG	1.000	0.002	0.005	0.131	0.004	0.007	0.146	0.005	0.010
IGR	1.000	0.003	0.007	0.132	0.004	0.007	0.146	0.005	0.010
MI	0.437	0.213	0.287	0.017	0.029	0.022	0.048	0.102	0.065
NMI	0.714	0.015	0.029	0.086	0.017	0.028	0.132	0.030	0.049
LL	1.000	0.002	0.005	0.132	0.004	0.007	0.146	0.005	0.010

CHI	**1.000**	0.002	0.004	**0.733**	0.002	0.005	**0.813**	0.003	0.006
Meta	1.000	0.002	0.005	0.652	0.003	0.007	0.680	0.004	0.008
tf-idf	0.719	0.010	0.020	0.211	0.003	0.005	0.468	0.007	0.013

Table 11: Score-thresholding results of PoS-tagged word patterns along with prepositions

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Threshold score > 0.01									
IG	0.447	0.675	0.538	0.018	0.050	0.027	0.050	0.102	0.067
IGR	0.447	0.622	0.520	0.019	0.050	0.027	0.050	0.109	0.069
MI	0.452	0.700	0.550	0.013	0.070	0.021	0.039	0.274	0.069
NMI	0.450	0.659	0.535	0.013	0.067	0.022	0.040	0.254	0.070
LL	0.447	0.622	0.520	0.019	0.050	0.027	0.050	0.109	0.068
CHI	0.452	0.345	0.391	0.026	0.039	0.031	0.081	0.056	0.066
Meta	0.448	0.612	0.517	0.014	0.059	0.022	0.042	0.225	0.071
tf-idf	0.409	0.609	0.489	0.016	0.068	0.026	0.045	0.244	0.076
Threshold score > 0.02									
IG	0.453	0.412	0.432	0.030	0.035	0.032	0.054	0.071	0.062
IGR	0.447	0.444	0.446	0.030	0.035	0.032	0.054	0.072	0.062
MI	0.449	0.634	0.525	0.013	0.063	0.021	0.040	0.250	0.069
NMI	0.450	0.532	0.487	0.015	0.061	0.024	0.042	0.221	0.071
LL	0.447	0.444	0.446	0.030	0.035	0.032	0.054	0.072	0.062
CHI	0.452	0.238	0.312	0.086	0.024	0.038	0.114	0.035	0.054
Meta	0.448	0.437	0.442	0.017	0.048	0.025	0.044	0.168	0.070
tf-idf	0.411	0.554	0.472	0.018	0.059	0.028	0.049	0.195	0.078
Threshold score > 0.03									
IG	0.453	0.412	0.432	0.035	0.023	0.028	0.054	0.049	0.051
IGR	0.454	0.356	0.399	0.034	0.024	0.028	0.054	0.050	0.052
MI	0.447	0.558	0.496	0.014	0.061	0.022	0.041	0.228	0.070
NMI	0.448	0.443	0.445	0.015	0.056	0.024	0.043	0.194	0.071
LL	0.454	0.356	0.399	0.035	0.024	0.028	0.053	0.050	0.051
CHI	0.451	0.206	0.283	0.156	0.020	0.036	0.194	0.026	0.046
Meta	0.451	0.408	0.428	0.020	0.044	0.028	0.049	0.141	0.073
tf-idf	0.426	0.424	0.425	0.020	0.051	0.029	0.054	0.153	0.080

Threshold score > 0.04									
IG	0.457	0.252	0.325	0.040	0.021	0.028	0.059	0.039	0.047
IGR	0.458	0.253	0.326	0.037	0.021	0.027	0.058	0.039	0.047
MI	0.448	0.527	0.484	0.014	0.058	0.023	0.042	0.216	0.070
NMI	0.446	0.428	0.437	0.016	0.052	0.024	0.045	0.183	0.072
LL	0.458	0.253	0.326	0.037	0.021	0.027	0.058	0.039	0.047
CHI	0.447	0.203	0.279	0.182	0.017	0.032	0.228	0.021	0.039
Meta	0.451	0.323	0.376	0.024	0.039	0.030	0.058	0.121	0.079
tf-idf	0.411	0.297	0.345	0.026	0.042	0.032	0.066	0.121	0.085
Threshold score > 0.05									
IG	0.699	0.039	0.074	0.048	0.018	0.027	0.063	0.034	0.044
IGR	0.459	0.221	0.299	0.046	0.018	0.026	0.063	0.034	0.044
MI	0.449	0.490	0.468	0.014	0.055	0.023	0.043	0.205	0.071
NMI	0.444	0.405	0.424	0.016	0.049	0.024	0.045	0.176	0.071
LL	0.459	0.221	0.299	0.048	0.018	0.027	0.063	0.034	0.044
CHI	0.444	0.200	0.276	0.198	0.015	0.027	0.242	0.018	0.034
Meta	0.447	0.293	0.354	0.026	0.039	0.031	0.060	0.118	0.079
tf-idf	0.440	0.183	0.258	0.030	0.030	0.030	0.073	0.097	0.084
Threshold score > 0.06									
IG	0.699	0.039	0.074	0.050	0.016	0.025	0.066	0.026	0.038
IGR	0.698	0.039	0.074	0.050	0.016	0.025	0.066	0.026	0.038
MI	0.445	0.436	0.440	0.014	0.052	0.023	0.043	0.189	0.070
NMI	0.449	0.396	0.421	0.016	0.047	0.024	0.046	0.167	0.072
LL	0.698	0.039	0.074	0.050	0.016	0.025	0.066	0.026	0.038
CHI	0.741	0.019	0.038	0.230	0.013	0.025	0.248	0.017	0.031
Meta	0.452	0.244	0.317	0.026	0.037	0.030	0.101	0.051	0.068
tf-idf	0.464	0.129	0.202	0.048	0.024	0.032	0.116	0.074	0.090
Threshold score > 0.07									
IG	0.689	0.034	0.065	0.056	0.015	0.023	0.068	0.024	0.036
IGR	0.697	0.035	0.067	0.056	0.015	0.023	0.068	0.024	0.036
MI	0.443	0.423	0.433	0.015	0.050	0.023	0.043	0.183	0.070
NMI	0.451	0.354	0.397	0.017	0.046	0.025	0.047	0.159	0.072
LL	0.697	0.035	0.067	0.056	0.015	0.023	0.068	0.024	0.036
CHI	0.741	0.019	0.038	0.351	0.010	0.020	0.254	0.015	0.028
Meta	0.452	0.214	0.290	0.076	0.027	0.040	0.105	0.048	0.066

tf-idf	0.483	0.090	0.151	0.068	0.018	0.028	0.124	0.057	0.078
Threshold score > 0.08									
IG	0.689	0.034	0.065	0.077	0.014	0.024	0.071	0.021	0.032
IGR	0.675	0.032	0.061	0.077	0.014	0.024	0.072	0.021	0.033
MI	0.443	0.420	0.431	0.015	0.049	0.022	0.044	0.179	0.070
NMI	0.449	0.340	0.387	0.018	0.045	0.025	0.047	0.151	0.072
LL	0.675	0.032	0.061	0.077	0.014	0.024	0.071	0.021	0.032
CHI	**0.763**	0.018	0.035	**0.366**	0.009	0.018	**0.262**	0.013	0.026
Meta	0.453	0.208	0.285	0.076	0.025	0.038	0.107	0.043	0.062
tf-idf	0.529	0.056	0.102	0.085	0.011	0.020	0.178	0.036	0.060
Threshold score > 0.09									
IG	0.664	0.029	0.056	0.068	0.012	0.020	0.071	0.019	0.030
IGR	0.664	0.029	0.056	0.068	0.012	0.020	0.071	0.019	0.030
MI	0.444	0.413	0.428	0.015	0.048	0.023	0.044	0.174	0.070
NMI	0.448	0.338	0.385	0.019	0.044	0.026	0.048	0.143	0.072
LL	0.664	0.029	0.056	0.068	0.012	0.020	0.071	0.019	0.030
CHI	**0.760**	0.017	0.034	**0.405**	0.009	0.018	**0.243**	0.010	0.019
Meta	0.451	0.206	0.283	0.095	0.023	0.037	0.118	0.040	0.060
tf-idf	0.569	0.035	0.066	0.097	0.007	0.014	0.195	0.025	0.044
Threshold score > 0.1									
IG	0.662	0.026	0.051	0.067	0.011	0.019	0.091	0.017	0.029
IGR	0.667	0.027	0.052	0.067	0.011	0.019	0.091	0.017	0.029
MI	0.444	0.403	0.423	0.015	0.046	0.022	0.044	0.170	0.070
NMI	0.447	0.309	0.365	0.018	0.043	0.026	0.048	0.142	0.072
LL	0.667	0.027	0.052	0.067	0.011	0.019	0.091	0.017	0.029
CHI	0.757	0.016	0.032	**0.443**	0.008	0.016	**0.526**	0.009	0.018
Meta	**0.774**	0.025	0.049	0.099	0.023	0.037	0.121	0.038	0.058
tf-idf	0.608	0.027	0.051	0.139	0.006	0.011	0.246	0.017	0.032
Threshold score > 0.2									
IG	**0.944**	0.005	0.010	0.202	0.006	0.012	0.153	0.008	0.015
IGR	0.944	0.005	0.010	0.202	0.006	0.012	0.153	0.008	0.015
MI	0.443	0.303	0.359	0.015	0.038	0.022	0.047	0.135	0.069
NMI	0.445	0.201	0.277	0.026	0.034	0.029	0.063	0.108	0.080
LL	0.944	0.005	0.010	0.202	0.006	0.012	0.153	0.008	0.015
CHI	0.909	0.003	0.006	**0.680**	0.005	0.010	**0.692**	0.006	0.011

Meta	0.871	0.008	0.016	0.214	0.011	0.021	0.227	0.013	0.024
tf-idf	0.654	0.016	0.032	0.216	0.003	0.007	0.283	0.012	0.023
Threshold score > 0.3									
IG	1.000	0.002	0.005	0.170	0.005	0.010	0.170	0.006	0.011
IGR	1.000	0.002	0.005	0.170	0.005	0.010	0.170	0.006	0.011
MI	0.438	0.274	0.337	0.017	0.035	0.023	0.047	0.124	0.068
NMI	0.725	0.018	0.035	0.091	0.021	0.034	0.138	0.038	0.060
LL	1.000	0.002	0.005	0.170	0.005	0.010	0.170	0.006	0.011
CHI	**1.000**	0.002	0.004	**0.750**	0.004	0.007	**0.706**	0.004	0.007
Meta	0.917	0.003	0.007	0.242	0.007	0.013	0.233	0.008	0.016
tf-idf	0.690	0.009	0.018	0.296	0.002	0.005	0.339	0.006	0.013
Threshold score > 0.4									
IG	1.000	0.002	0.003	0.163	0.004	0.008	0.159	0.005	0.010
IGR	1.000	0.002	0.003	0.163	0.004	0.008	0.159	0.005	0.010
MI	0.441	0.226	0.299	0.019	0.032	0.024	0.051	0.112	0.070
NMI	0.735	0.015	0.030	0.109	0.020	0.033	0.150	0.032	0.053
LL	1.000	0.002	0.003	0.163	0.004	0.008	0.159	0.005	0.010
CHI	**1.000**	0.001	0.002	**0.833**	0.003	0.006	**0.846**	0.003	0.007
Meta	1.000	0.002	0.004	0.750	0.004	0.007	0.706	0.004	0.007
tf-idf	0.704	0.006	0.012	0.417	0.002	0.003	0.417	0.003	0.006

Table 12: Score-thresholding results of verb-centred word patterns along with preposition

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Top 100 Ranked Patterns									
IG	0.770	0.025	0.049	0.210	0.007	0.013	0.210	0.007	0.013
IGR	0.770	0.025	0.049	0.210	0.007	0.013	0.210	0.007	0.013
MI	0.560	0.018	0.036	0.020	0.001	0.001	0.190	0.006	0.012
NMI	0.940	0.031	0.060	0.410	0.014	0.026	0.510	0.017	0.033
LL	0.770	0.025	0.049	0.210	0.007	0.013	0.210	0.007	0.013
CHI	**0.960**	0.032	0.061	**0.420**	0.014	0.027	**0.510**	0.017	0.033
Meta	0.900	0.030	0.057	0.350	0.012	0.022	0.430	0.014	0.027
tf-idf	0.920	0.030	0.059	0.390	0.013	0.025	0.460	0.015	0.029
Top 200 Ranked Patterns									

IG	0.800	0.053	0.099	0.135	0.009	0.017	0.240	0.016	0.030
IGR	0.800	0.053	0.099	0.135	0.009	0.017	0.235	0.016	0.029
MI	0.560	0.037	0.069	0.045	0.003	0.006	0.160	0.011	0.020
NMI	0.815	0.054	0.101	0.330	0.022	0.041	0.445	0.029	0.055
LL	0.800	0.053	0.099	0.135	0.009	0.017	0.235	0.016	0.029
CHI	0.815	0.054	0.101	**0.395**	0.026	0.049	**0.445**	0.030	0.056
Meta	**0.830**	0.055	0.103	0.365	0.024	0.045	0.425	0.028	0.053
tf-idf	0.810	0.053	0.100	0.360	0.024	0.045	0.430	0.028	0.053
Top 300 Ranked Patterns									
IG	0.780	0.077	0.140	0.167	0.016	0.030	0.210	0.021	0.038
IGR	0.787	0.078	0.142	0.167	0.016	0.030	0.210	0.021	0.038
MI	0.540	0.053	0.097	0.037	0.004	0.007	0.163	0.016	0.029
NMI	0.707	0.070	0.127	0.277	0.027	0.050	0.353	0.035	0.064
LL	**0.790**	0.078	0.142	0.167	0.016	0.030	0.210	0.021	0.038
CHI	0.710	0.070	0.128	**0.380**	0.038	0.068	**0.440**	0.044	0.079
Meta	0.740	0.073	0.133	0.310	0.031	0.056	0.377	0.037	0.068
tf-idf	0.707	0.070	0.128	0.337	0.033	0.061	0.387	0.038	0.070

Table 13: Rank-thresholding results of adapted linked chain dependency patterns

	GENIA			**WEB**			**GENIA + WEB**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Threshold score > 0.01									
IG	0.748	0.107	0.187	0.150	0.073	0.098	0.223	0.145	0.176
IGR	0.748	0.107	0.187	0.153	0.076	0.101	0.225	0.144	0.176
MI	0.567	0.816	0.669	0.048	0.190	0.077	0.161	0.822	0.269
NMI	0.566	0.767	0.651	0.049	0.179	0.077	0.163	0.771	0.268
LL	0.748	0.107	0.187	0.151	0.077	0.102	0.225	0.144	0.176
CHI	0.577	0.529	0.552	0.191	0.059	0.090	0.263	0.099	0.144
Meta	0.571	0.643	0.605	0.051	0.144	0.076	0.161	0.596	0.253
tf-idf	0.553	0.575	0.564	0.054	0.157	0.080	0.176	0.527	0.264
Threshold score > 0.02									
IG	0.796	0.051	0.097	0.199	0.054	0.085	0.263	0.094	0.138
IGR	0.796	0.051	0.097	0.199	0.054	0.085	0.264	0.093	0.137
MI	0.566	0.744	0.643	0.048	0.174	0.076	0.162	0.758	0.267

NMI	0.570	0.706	0.631	0.051	0.162	0.078	0.165	0.687	0.266
LL	0.796	0.051	0.097	0.199	0.054	0.085	0.264	0.093	0.137
CHI	0.591	0.243	0.344	0.327	0.042	0.074	0.330	0.064	0.107
Meta	0.569	0.547	0.558	0.053	0.120	0.073	0.163	0.496	0.245
tf-idf	0.557	0.532	0.544	0.057	0.131	0.079	0.184	0.457	0.263
Threshold score > 0.03									
IG	0.785	0.035	0.067	0.220	0.047	0.078	0.263	0.074	0.116
IGR	0.785	0.035	0.067	0.219	0.047	0.077	0.263	0.074	0.116
MI	0.566	0.711	0.631	0.048	0.165	0.074	0.164	0.734	0.268
NMI	0.568	0.663	0.612	0.050	0.148	0.074	0.165	0.646	0.263
LL	0.785	0.035	0.067	0.220	0.047	0.078	0.263	0.074	0.116
CHI	0.613	0.146	0.236	0.414	0.033	0.061	0.426	0.040	0.073
Meta	0.577	0.355	0.439	0.056	0.105	0.073	0.164	0.403	0.233
tf-idf	0.567	0.491	0.526	0.058	0.088	0.070	0.225	0.337	0.270
Threshold score > 0.04									
IG	0.784	0.025	0.049	0.203	0.033	0.056	0.259	0.061	0.098
IGR	0.786	0.025	0.049	0.203	0.033	0.056	0.260	0.060	0.098
MI	0.566	0.681	0.618	0.048	0.150	0.073	0.163	0.662	0.261
NMI	0.569	0.620	0.593	0.050	0.140	0.074	0.164	0.608	0.258
LL	0.786	0.025	0.049	0.203	0.033	0.056	0.260	0.060	0.098
CHI	0.604	0.139	0.226	0.429	0.024	0.045	0.443	0.033	0.062
Meta	0.586	0.237	0.337	0.106	0.079	0.090	0.200	0.189	0.194
tf-idf	0.575	0.412	0.480	0.115	0.080	0.094	0.246	0.254	0.250
Threshold score > 0.05									
IG	0.727	0.018	0.036	0.209	0.029	0.051	0.268	0.050	0.085
IGR	0.727	0.018	0.036	0.209	0.029	0.051	0.268	0.050	0.085
MI	0.566	0.658	0.608	0.048	0.144	0.072	0.163	0.641	0.260
NMI	0.567	0.598	0.582	0.050	0.130	0.072	0.161	0.550	0.249
LL	0.727	0.018	0.036	0.209	0.029	0.051	0.268	0.050	0.085
CHI	0.595	0.130	0.214	0.418	0.019	0.037	0.456	0.027	0.052
Meta	0.604	0.145	0.234	0.103	0.072	0.085	0.195	0.176	0.185
tf-idf	0.581	0.363	0.446	0.121	0.069	0.088	0.297	0.166	0.213
Threshold score > 0.06									
IG	0.685	0.012	0.024	0.198	0.026	0.046	0.273	0.045	0.078
IGR	0.685	0.012	0.024	0.198	0.026	0.046	0.273	0.045	0.078

MI	0.564	0.646	0.602	0.047	0.139	0.071	0.163	0.627	0.259
NMI	0.565	0.563	0.564	0.051	0.122	0.072	0.162	0.522	0.247
LL	0.685	0.012	0.024	0.198	0.026	0.046	0.273	0.045	0.078
CHI	0.865	0.051	0.096	0.446	0.016	0.032	0.507	0.023	0.044
Meta	0.600	0.137	0.223	0.139	0.062	0.086	0.222	0.125	0.160
tf-idf	0.631	0.287	0.395	0.152	0.062	0.088	0.378	0.131	0.195
Threshold score > 0.07									
IG	0.630	0.010	0.019	0.191	0.022	0.040	0.265	0.042	0.073
IGR	0.630	0.010	0.019	0.191	0.022	0.040	0.264	0.042	0.072
MI	0.565	0.620	0.591	0.047	0.133	0.069	0.162	0.601	0.256
NMI	0.563	0.537	0.550	0.050	0.118	0.070	0.162	0.508	0.245
LL	0.630	0.010	0.019	0.191	0.022	0.040	0.264	0.042	0.072
CHI	0.871	0.049	0.092	0.430	0.013	0.026	0.516	0.021	0.040
Meta	0.594	0.130	0.213	0.142	0.058	0.082	0.222	0.114	0.151
tf-idf	0.713	0.203	0.316	0.170	0.053	0.081	0.417	0.084	0.140
Threshold score > 0.08									
IG	0.758	0.008	0.016	0.181	0.020	0.036	0.255	0.039	0.067
IGR	0.758	0.008	0.016	0.181	0.020	0.036	0.255	0.039	0.067
MI	0.565	0.607	0.585	0.047	0.131	0.070	0.162	0.588	0.254
NMI	0.565	0.526	0.545	0.050	0.113	0.069	0.162	0.486	0.243
LL	0.758	0.008	0.016	0.181	0.020	0.036	0.255	0.039	0.067
CHI	**0.906**	0.038	0.073	**0.422**	0.012	0.022	**0.528**	0.019	0.036
Meta	0.795	0.060	0.112	0.187	0.049	0.077	0.157	0.088	0.113
tf-idf	0.823	0.124	0.216	0.206	0.046	0.075	0.439	0.050	0.090
Threshold score > 0.09									
IG	0.733	0.007	0.014	0.167	0.016	0.030	0.259	0.036	0.064
IGR	0.733	0.007	0.014	0.168	0.016	0.030	0.259	0.036	0.064
MI	0.563	0.593	0.578	0.047	0.129	0.069	0.160	0.572	0.250
NMI	0.572	0.507	0.538	0.051	0.109	0.070	0.162	0.463	0.240
LL	0.733	0.007	0.014	0.167	0.016	0.030	0.259	0.036	0.064
CHI	**0.900**	0.036	0.069	**0.667**	0.008	0.016	**0.515**	0.016	0.032
Meta	0.860	0.048	0.092	0.217	0.046	0.076	0.259	0.080	0.122
tf-idf	0.854	0.058	0.109	0.311	0.032	0.057	0.443	0.028	0.053
Threshold score > 0.1									
IG	0.704	0.006	0.012	0.174	0.015	0.027	0.252	0.034	0.060

IGR	0.704	0.006	0.012	0.174	0.015	0.027	0.252	0.034	0.060
MI	0.564	0.588	0.576	0.048	0.122	0.069	0.159	0.554	0.248
NMI	0.569	0.483	0.523	0.050	0.106	0.068	0.162	0.460	0.240
LL	0.704	0.006	0.012	0.174	0.015	0.027	0.252	0.034	0.060
CHI	**0.898**	0.035	0.067	**0.714**	0.007	0.013	**0.500**	0.015	0.029
Meta	0.856	0.047	0.089	0.207	0.042	0.070	0.251	0.075	0.115
tf-idf	0.866	0.045	0.085	0.371	0.021	0.039	0.476	0.022	0.043
Threshold score > 0.2									
IG	0.571	0.003	0.005	0.159	0.004	0.007	0.210	0.007	0.013
IGR	0.571	0.003	0.005	0.159	0.004	0.007	0.210	0.007	0.013
MI	0.566	0.473	0.515	0.044	0.090	0.059	0.157	0.456	0.234
NMI	0.600	0.133	0.218	0.105	0.064	0.079	0.200	0.155	0.175
LL	0.571	0.003	0.005	0.159	0.004	0.007	0.210	0.007	0.013
CHI	**1.000**	0.015	0.029	**0.800**	0.003	0.005	**0.917**	0.004	0.007
Meta	1.000	0.013	0.025	0.337	0.011	0.020	0.434	0.021	0.040
tf-idf	0.879	0.019	0.037	0.443	0.013	0.025	0.737	0.009	0.018
Threshold score > 0.3									
IG	1.000	0.001	0.001	0.195	0.003	0.005	0.211	0.005	0.010
IGR	1.000	0.001	0.001	0.195	0.003	0.005	0.211	0.005	0.010
MI	0.562	0.320	0.408	0.040	0.074	0.052	0.154	0.380	0.220
NMI	0.812	0.055	0.104	0.141	0.047	0.070	0.230	0.090	0.130
LL	1.000	0.001	0.001	0.195	0.003	0.005	0.211	0.005	0.010
CHI	**1.000**	0.009	0.018	**1.000**	0.001	0.002	**1.000**	0.002	0.004
Meta	1.000	0.004	0.008	0.302	0.005	0.010	0.364	0.008	0.015
tf-idf	0.895	0.011	0.022	0.656	0.010	0.020	0.842	0.005	0.010
Threshold score > 0.4									
IG	1.000	0.001	0.001	0.154	0.002	0.004	0.236	0.004	0.008
IGR	1.000	0.001	0.001	0.154	0.002	0.004	0.236	0.004	0.008
MI	0.569	0.209	0.306	0.040	0.064	0.049	0.154	0.329	0.209
NMI	0.939	0.031	0.059	0.203	0.036	0.061	0.281	0.057	0.095
LL	1.000	0.001	0.001	0.154	0.002	0.004	0.236	0.004	0.008
CHI	**1.000**	0.005	0.010	**1.000**	0.001	0.001	**1.000**	0.001	0.002
Meta	1.000	0.001	0.003	0.154	0.002	0.004	0.286	0.005	0.010
tf-idf	0.941	0.005	0.010	0.800	0.004	0.008	0.917	0.004	0.007

Table 14: Score-thresholding results of adapted linked chain dependency patterns

Bibliography

Abell, M., Bauder, D. & Simmons, T. (2004). Universally designed online assessment: Implications for the future. *Information Technology and Disability*, 10(1).

Afzal, N., Mitkov, R. & Farzindar, A. (2011). Unsupervised relation extraction using dependency trees for automatic generation of multiple-choice questions. In *Proceedings of the C. Butz and P. Lingras (Eds.): Canadian Artificial Intelligence, LNAI 6657*. Newfoundland and Labrador, Canada: Springer, Heidelberg, pp. 32-43.

Afzal, N. & Pekar, V. (2009). Unsupervised relation extraction for automatic generation of multiple-choice questions. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP-2009)*. Borovets, Bulgaria, pp. 1-5.

Agichtein, E. & Ganti, V. (2004). Mining reference tables for automatic text segmentation. In *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-04)*. Seattle, USA: ACM Press, pp. 20-29.

Agichtein, E. & Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM conference on Digital libraries*. ACM, pp. 85-94.

Aldabe, I. & Maritxalar, M. (2010). Automatic distractors generation for domain specific texts. In *Proceedings of the 7th International Conference on Advances in Natural Language Processing (IceTAL-2010)*. pp. 27-38.

Alfonseca, E. & Manandhar, S. (2002). An unsupervised method for general named entity recognition and automated concept discovery. In *Proceedings of the 1st International Conference on General WordNet*. pp. 1-9.

Ananiadou, S. & McNaught, J. eds. (2006). *Text Mining for Biology and Biomedicine*, Artech House.

Ando, R.K. & Zhang, T. (2005). A high-performance semi-supervised learning method for text chunking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL-05)*. Association for Computational Linguistics, pp. 1-9.

Andrade, M.A. & Valencia, A. (1998). Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. *Bioinformatics*, 14(7), pp. 600-607.

Asahara, M. & Matsumoto, Y. (2003). Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL- 03)*. pp. 8-15.

Bach, N. & Badaskar, S. (2007). A survey on relation extraction. Language Technologies Institute, Carnegie Mellon University.

Ball, S., Barber, C., Buckel, L., Cooke, S., Gluc, E., Mole, J. & Sutherland, A. (2003). Inclusive learning and teaching: ILT for disabled learners. *Becta Ferl and JISC TechDis*.

Barzilay, R. & Lapata, M. (2005). Collective content selection for concept-to-text generation. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-05)*. pp. 331-338.

Basili, R., Pazienza, M. & Vindigni, M. (2000). Corpus-driven learning of event recognition rules. In *Proceedings of the ECAI 2001 Workshop on Machine Learning for Information Extraction*. August, pp. 20-25.

Becker, W.E. & Watts, M. (2001). Teaching methods in U.S. undergraduate economics courses. *Journal of Economic Education*, pp.269-280.

Becker, W.E. & Johnston, C. (1999). The relationship between multiple choice and essay response questions in assessing economics understanding. *Economic Record*, 75(4), pp. 348-357.

Benton, M., Tremaine, M. & Scher, J. (2004). Computer aids for designing effective multiple choice questions. In *Proceedings of Americas Conference on Information Systems (AMCIS).*

Bikel, D.M., Schwartz, R. & Weischedel, R.M. (1999). An algorithm that learns what's in a name. *Machine Learning*, 34(1), pp. 211-231.

Bikel, D.M., Miller, S., Schwartz, R. & Weischedel, R. (1998). Nymble: A high-performance learning name-finder. In *Proceedings of the 5th Conference on Applied Natural Language Processing*. Association for Computational Linguistics, p. 8.

Blaschke, C., Andrade, M., Ouzounis, C. & Valencia, A. (1999). Automatic extraction of biological information from scientific text: protein-protein interactions. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB)*. pp. 60-67.

Borkar, V., Deshmukh, K. & Sarawagi, S. (2001). Automatic segmentation of text into structured records. In *Proceedings of ACM SIGMOD International Conference of Management of Data*. Santa Barabara, USA, pp. 175-186.

Borthwick, A., Sterling, J., Agichtein, E. & Grishman, R. (1998). NYU: Description of the MENE named entity system as used in MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Virginia, USA.

Boytcheva, S., Nikolova, I., Paskaleva, E., Angelova, G., Tcharaktchiev, D. & Dimitrova, N. (2009). Extraction and exploration of correlations in patient status data. In *Proceedings of RANLP 2009 Workshop: Biomedical Information Extraction*. Borovets, Bulgaria, pp. 1-7.

Brin, S. (1998). Extracting patterns and relations from the World Wide Web. In *Proceedings of the International Workshop on World Wide Web and Databases*. pp. 172–183.

Brown, J.C., Frishkoff, G. & Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-05).* October, Vancouver, pp. 819-826.

Bunescu, R. & Mooney, R. (2007). Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07).* Prague, Czech Republic.

Bunescu, R. & Mooney, R. (2006). Subsequence kernels for relation extraction. In Y. Weiss, B. Schölkopf, & J. Platt, eds. *Advances in Neural Information Processing Systems 18*. MIT Press, p. 171.

Bunescu, R., Ge, R., Kate, R., Marcotte, M., Mooney, R., Ramani, A. & Wong, Y. (2005). Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2), pp. 139-155.

Cai, Y., Dong, X., Halevy, A., Liu, J. & Madhavan, J. (2005). Personal information management with SEMEX. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD-05).* ACM Press, pp. 921-923.

Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E. & Li, H. (2008). Context-aware query suggestion by mining click-through and session data. In *Proceedings of KDD-08*. pp. 875-883.

Caraballo, S.A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*. pp. 120-126.

Carreras, X., Màrquez, L. & Padró, L. (2003). A simple named entity extractor using AdaBoost. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*. Edmonton, Canada, pp. 152-155.

Carter, J., Ala-Mutka, K., Fuller, U., Dick, M., English, J., Fone, W. & Sheard, J. (2003). How shall we assess this? *ACM SIGCSE Bulletin*, 35(4), pp. 107-123.

Català, N. (2003). *Acquiring Information Extraction patterns from unannotated corpora*. PhD thesis, Technical University of Catalonia.

Català, N., Castell, N. & Martin, M. (2000). ESSENCE: A portable methodology for acquiring Information Extraction patterns. In *Proceedings of 14th European Conference on Artificial Intelligence (ECAI-2000).* pp. 411-415.

Chakrabarti, S., Mirchandani, J. & Nandi, A. (2005). SPIN: Searching Personal Information Networks. In *Proceedings of Annual ACM Conference on Research and Development in Information Retrieval*. p. 674.

Chang, W., Pantel, P., Popescu, A.M. & Gabrilovich, E. (2009). Towards intent-driven bidterm suggestion. In *Proceedings of the 18th International Conference on World Wide Web (WWW-09).* pp. 1093-1094.

Chen, C.Y., Liou, H.C. & Chang, J.S. (2006). FAST: An automatic generation system for grammar tests. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*. Association for Computational Linguistics, pp. 1–4.

Chen, L., Liu, H. & Friedman, C. (2005). Gene name ambiguity of eukaryotic nomenclatures. *Bioinformatics*, 21(2), pp. 248-256.

Chen, W., Aist, G. & Mostow, J. (2009). Generating questions automatically from informational text. In *Proceedings of the 2nd Workshop on Question Generation.* Brighton, UK, pp. 17-24.

Chieu, H.L. & Ng, H.T. (2003). Named entity recognition with a maximum entropy approach. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*. Edmonton, Canada, pp. 160-163.

Chinchor, N. (1998). MUC-7 named entity task definition, version 3.5. In *Proceedings of the Seventh Message Understanding Conference (MUC-7).*

Cohen, A.M. & Hersh, W.R. (2005). A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1), pp. 57-71.

Cohen, J. (1968). Weighted Kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4), pp. 213-220.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), pp. 37-46.

Collins, M. & Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. pp. 100-110.

Corney, D., Buxton, B., Langdon, W. & Jones, D. (2004). BioRAT: Extracting biological information from full-length papers. *Bioinformatics (Oxford, England)*, 20(17), pp. 3206-3213.

Cover, T. & Thomas, J. (1991). *Elements of Information Theory*, New York, USA: Wiley.

Cucchiarelli, A. & Velardi, P. (2001). Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Computational Linguistics*, 27(1), pp. 123-131.

Culotta, A. & Sorensen, J. (2004). Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL-04)*. Barcelona, Spain: Association for Computational Linguistics, pp. 423-429.

Cunningham, H., Maynard, D., Bontcheva, K. & Tablan, V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 168-175.

Cutrell, E. & Dumais, S.T. (2006). Exploring personal information. *Communications of the ACM*, 49(4), pp. 50-51.

Dagan, I. (2000). Contextual word similarity. In R. Dale, H. Moisl, & H. Somers, eds. *Handbook of Natural Language Processing*. Marcel Dekker Inc, pp. 459-476.

Dagan, I., Lee, L. & Pereira, F. (1999). Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3), pp. 43-69.

Dagan, I., Lee, L. & Pereira, F. (1997). Similarity-based methods for word sense disambiguation. In *Proceedings of the 35th Annual Meeting on Association for Computational Linguistics*. Madrid, Spain, pp. 56-63.

Das, R. & Elikkottil, A. (2010). Auto-summarizer to aid a Q/A system. *International Journal of Computer Applications*, 1(1), pp. 113-117.

Dhillon, I.S., Mallela, S. & Kumar, R. (2002). Enhanced word clustering for hierarchical text classification. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)*. pp. 191-200.

Downey, D., Etzioni, O. & Soderland, S. (2005). A probabilistic model of redundancy in information extraction. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*. pp. 1034-1041.

Dufresne, R.J., Leonard, W.J. & Gerace, W.J. (2002). Making sense of students' answers to multiple-choice questions. *The Physics Teacher*, 40(3), pp. 174-180.

Eichler, K., Hemsen, H. & Neumann, G. (2008). Unsupervised relation extraction from web documents. In *Proceedings of the 6th International Language Resources and Evaluation (LREC-08).* Marrakech, Morocco: European Language Resources Association (ELRA).

Erk, K. (2007). A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL-07).* Prague, Czech Republic, pp. 216-223.

Erkan, G., Ozgur, A. & Radev, D.R. (2007). Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL).* Prague, Czech Republic, pp. 228-237.

Etzioni, O., Banko, M., Soderland, S. & Weld, D.S. (2008). Open Information Extraction from the Web. *Communications of the ACM*, 51(12), pp. 68-74.

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D. & Yates, A. (2005). Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, 165(1), pp. 91-134.

Evans, R. (2003). A framework for named entity recognition in the open domain. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP-2003).* Borovets, Bulgaria, pp. 137-144.

Farzindar, A. & Lapalme, G. (2004). LetSum, an automatic legal text summarizing system. In T. F. Gordon, ed. *Legal Knowledge and Information Systems, JURIX 2004.* Berlin, Germany: IOS Press, pp. 11-18.

Feldman, R. (2006). Information Extraction - theory and practice. In *Proceedings of 23rd International Conference on Machine Learning (ICML).* Pittsburgh, Pennsylvania.

Finkel, J.R., Grenager, T. & Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 363-370.

Firth, J.R. (1957). A synopsis of linguistic theory 1930-1955. F. Palmer, ed. *Studies in Linguistic Analysis*, Special volume, pp. 1-32.

Florian, R. Han, B., Luo, X., Kambhatla, N. & Zitouni, I. (2007). IBM ACE-07 system description. In *Proceedings of NIST 2007 Automatic Content Extraction Evaluation*.

Frakes, B.W. & Baeze-Yates, R. (1992). *Information Retrieval, Data Structures and Algorithms*, Prentice Hall.

Fundel, K., Küffner, R. & Zimmer, R. (2007). RelEx- relation extraction using dependency parse trees. *Bioinformatics*, 23(3), pp. 365-371.

Fundel, K. & Zimmer, R. (2006). Gene and protein nomenclature in public databases. *BMC Bioinformatics*, 7, p.372.

Gates, D. (2008). Generating look-back strategy questions from expository texts. In *Proceedings of 1st Workshop on the Question Generation Shared Task and Evaluation Challenge*. Arlington, VA, pp. 1-3.

Goodrich, C.H. (1977). Distractor efficiency in foreign language testing. *TESOL Quarterly*, 11(1), pp. 69-78.

Graesser, A.C., Chipman, P., Haynes, B.C. & Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4), pp. 612-618.

Graesser, A.C. & Person, N.K. (1994). Question asking during tutoring. *American Educational Research Journal*, 31(1), pp. 104-137.

Grover, C., Lascarides, A. & Lapata, M. (2005). A comparison of parsing technologies for the biomedical domain. *Natural Language Engineering*, 11(1), pp. 27-65.

Grefenstette, G. (1994). Explorations in automatic thesaurus discovery. *Kluwer, Bostin: The Springer International Series in Engineering and Computer Science, Vol. 278.*

Greenwood, M.A., Stevenson, M., Guo, Y., Harkema, H. & Roberts, A. (2005). Automatically acquiring a linguistically motivated genic interaction extraction system. In *Proceedings of the 4th Learning Language in Logic Workshop (LLL-05).* Bonn, Germany, pp. 1-7.

Grishman, R., Huttunen, S. & Yangarber, R. (2002). Information Extraction for enhanced access to disease outbreak reports. *Journal of Biomedical Informatics*, 35(4), pp. 236-246.

Grishman, R. & Sundheim, B. (1996). Message Understanding Conference-6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics*. Copenhagen, Denmark, pp. 466-471.

Gronlund, N. (1982). *Constructing Achievement Tests*. New York, USA: Prentice Hall.

Ha, L.A. (2007). *Advances in automatic terminology processing: Methodology and application in focus*. PhD thesis. University of Wolverhampton.

Haladyna, T., Downing, S. & Rodriguez, M. (2002). A review of multiple-choice item-writing guidelines for classroom assessment. *Applied Measurement in Education*, 15(3), pp. 309-333.

Harabagiu, S. & Maiorano, S. (2000). Acquisition of linguistic patterns for knowledge-based information extraction. In *Proceedings of LREC-2000.* Athens, Greece.

Harkema, H., Setzer, A., Gaizauskas, R., Hepple, M., Power, R. & Rogers, J. (2005). Mining and modelling temporal clinical data. In *Proceedings of the 4th UK e-Science All Hands Meeting*. Nottingham, UK.

Harris, Z. (1968). *Mathematical Structures of Language*. Interscience Publishers.

Harris, Z. (1954). Distributional structure. J. Katz, ed. *Word Journal of the International Linguistic Association*, 10(23), pp. 146-162.

Harshman, R.A. (1970). Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16(1), pp. 1-84.

Hasegawa, T., Sekine, S. & Grishman, R. (2004). Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.* Barcelona, Spain: Association for Computational Linguistics, pp. 415-422.

Hatzivassiloglou, V. (1996). Do we need linguistics when we have statistics? A comparative analysis of the contributions of linguistics cues to a statistical word grouping system. In J. Klavans & P. Resnik, eds. *The balancing act: Combining symbolic and statistical approaches to language*. MIT Press, pp. 67-94.

Hearst, M.A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics*. Association for Computational Linguistics, pp. 539-545.

Heng, J. (2008). *Improving Information Extraction and translation using component interactions*. PhD thesis. New York University.

Heng, J. & Grishman, R., (2006). Data selection in semi-supervised learning for name tagging. In *Proceedings of COLING/ACL 06 Workshop on Information Extraction Beyond Document*. Association for Computational Linguistics, pp. 48-55.

Higgins, D., Burstein, J., Marcu, D., & Gentile, C. (2004). Evaluating multiple aspects of coherence in student essays. In *Proceedings of the Human Language Technology Conference North American Chapter of the Association for Computational Linguistics*, pp. 185-192.

Hirschman, L. & Mani, I. (2003). Evaluation. In R. Mitkov, ed. *The Oxford Handbook of Computational Linguistics*. Oxford University Press, pp. 414-429.

Hodges, P.E., McKee, A.H., Davis, B.P., Payne, W.E. & Garrels, J.I. (1999). The Yeast Proteome Database (YPD): A model for the organization and presentation of genome-wide functional data. *Nucleic Acids Research*, 27(1), pp. 69-73.

Hoshino, A. & Nakagawa, H. (2007). Assisting cloze test making with a web application. In *Proceedings of Society for Information Technology Teacher Education International Conference 2007*. AACE, pp. 2807-2814.

Hoshino, A. & Nakagawa, H. (2005). A real-time multiple-choice question generation for language testing: a preliminary study. In *Proceedings of the 43rd ACL'05, Second Workshop on Building Educational Applications Using NLP*. pp. 17-20.

Huang, M., Zhu, X., Payan, G.D., Qu, K. & Li, M. (2004). Discovering patterns to extract protein-protein interactions from full texts. *Bioinformatics (Oxford, England)*, 20(18), pp. 3604-3612.

Huffman, S. (1996). Learning information extraction patterns from examples. *Connectionist Statistical and Symbolic Approaches to Learning for Natural Language Processing*, pp. 246-260.

Isaacs, G. (1994). Multiple choice testing: A guide to the writing of multiple choice tests and to their analysis. Campbell town: HERDSA.

Järvinen, T. (1994). Annotating 200 million words: the Bank of English project. In *Proceedings of the 15th International Conference on Computational Linguistics*. Kyoto, pp. 565-568.

Jiang, J.J. & Conrath, D.W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*. Taiwan, pp. 19-33.

Jurafsky, D. & Martin, J.H. (2008). Information Extraction. In *Speech and Language Processing*. Prentice Hall, pp. 725-764.

Kaiser, K. & Miksch, S. (2005). Information Extraction a survey. *Technology*, (May), p.32.

Kalady, S., Elikkottil, A. & Das, R. (2010). Natural language question generation using syntax and keywords. In *Boyer, Kristy Elizabeth and Piwek, Paul eds. (2010). Proceedings of QG2010: The 3rd Workshop on Question Generation*. Pittsburgh, Pennsylvania, pp. 1-11.

Karamanis, N., Ha, L.A. & Mitkov, R. (2006). Generating multiple-choice test items from medical text: A pilot study. In *Proceedings of the 4th International Natural Language Generation Conference*, (July), pp. 111-113.

Katrenko, S. & Adriaans, P. (2006). Learning relations from biomedical corpora using dependency tree levels. In *Proceedings of the 1st International Workshop on Knowledge Discovery and Emergent Complexity in Bioinformatics*. pp. 61-80.

Keller, F. & Lapata, M. (2003). Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3), pp. 459-484.

Kilgarriff, A. & Grefenstette, G. (2003). Introduction to the special issue on the Web as corpus. *Computational Linguistics*, 29(3), pp. 333-347.

Kilgarriff, A. (2001). Comparing corpora. *International Journal of Corpus Linguistics*, 6(1), pp. 1-37.

Kilgarriff, A. & Rose, T. (1998). Measures for corpus similarity and homogeneity. In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Granada, Spain, pp. 46-52.

Kilgarriff, A. (1997). Using word frequency lists to measure corpus homogeneity and similarity between corpora. In *Proceedings of 5th ACL-SIGDAT Workshop on very Large Corpora*. Beijing and Hong Kong, pp. 231–245.

Kim, J.-D., Ohta, T. & Tsujii, J. (2008). Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(1), p.10.

Kim, J.T. & Moldovan, I.D. (1995). Acquisition of linguistic patterns for knowledge-based information extraction. *IEEE Transactions on Knowledge and Data Engineering*, 7(5), pp. 713-724.

Klein, D. & Manning, C.D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 423-430.

Kleinberg, J.M. (2002). Bursty and hierarchical structure in streams. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)*. ACM Press, pp. 91-101.

Kulick, S., Bies, A., Liberman, M., Mandel, M., McDonald, R., Palmer, M., Schein, A. & Ungar, L. (2004). Integrated annotation for biomedical information extraction. In J. Pustejovsky & L. Hirschman, eds. *Proceedings of the HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*. Boston, Massachusetts, USA: Association for Computational Linguistics, pp. 61-68.

Kullback, S. & Leibler, R.A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1), pp. 79-86.

Kunichika, H. Urushima, M. Hirashima, T. & Takeuchi, A. (2002). A computational method of complexity of questions on contents of English sentences and its evaluation. In *Proceedings of the International Conference on Computers in Education (ICCE-02)*. Auckland, NZ, pp. 97-101.

Lapata, M., Keller, F. & McDonald, S. (2001). Evaluating smoothing algorithms against plausibility judgements. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*. Toulouse, France, pp. 346-353.

Lavelli, A., Califf, M., Ciravegna, F., Freitag, D., Giuliano, C., Kushmerik, N. & Romano, L. (2004). IE evaluation: Criticisms and recommendations. In *Proceedings of the AAAI Workshop on Adaptive Text Extraction and Mining*.

Leacock, C. & Chodorow, M. (2003). C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(no. 4), pp. 389-405.

Leacock, C. & Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, ed. *WordNet: An Electronic Lexical Database*. MIT Press, pp. 265-283.

Lee, L. (2001). On the effectiveness of the skew divergence for statistical language analysis. In *Proceedings of the Artificial Intelligence and Statistics*. pp. 65-72.

Lehnert, W., Cardie, C., Fisher, D., McCarthy, J., Riloff, E. & Soderland, S. (1992). University of Massachusetts: Description of the CIRCUS System as Used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*. Morgan Kaufmann, pp. 282-288.

Lehnert, W. (1990). Symbolic/subsymbolic sentence analysis: Exploiting the best of two worlds. In *Barnden, J. and Pollack, J., editors 1990, Advances in Connectionist and Neural Computation Theory*, 1, pp. 135–164.

Lesk, M. (1986). Automatic sense disambiguation: How to tell a pine cone from an ice cream cone. In *Proceedings of the SIGDOC Conference*. Toronto, Canada: Association for Computing Machinery, pp. 24-26.

Lin, D. & Pantel, P. (2001). DIRT- Discovery of Inference Rules from Text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. San Francisco, USA, pp. 323–328.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 768-774.

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*. Madrid, Spain, pp. 64-71.

Lin, D. (1991). MINIPAR: A minimalist parser. In *Maryland Linguistics Colloquium, University of Maryland.*

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1), pp. 145-151.

Linn, R.L. & Miller, M.D. (2005). *Measurement and Assessment in Teaching* 9th ed., Prentice Hall.

Lister, R. (2001). Objectives and objective assessment in CS1. *ACM SIGCSE Bulletin*, 33(1), pp. 292-296.

Lister, R. (2000). On blooming first year programming, and its blooming assessment. In *Proceedings of the Australasian Conference on Computing Education*. ACM, pp. 158-162.

Liu, B., Hu, M. & Cheng, J. (2005). Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*. ACM, pp. 342-351.

McDonald, R. (2005). Extracting relations from unstructured text. Technical report, Department of Computer and Information Science, University of Pennsylvania.

Manning, C.D., Raghavan, P. & Schütze, H. (2008). Evaluation in information retrieval. In *Introduction to Information Retrieval*. Cambridge, USA: Cambridge University Press, pp. 139-161.

Manning, C. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*, Cambridge, USA: The MIT Press.

Martin, E.P., Bremer, E., Guerin, G., DeSesa, M-C. & Jouve, O. (2004). Analysis of protein-protein interactions through biomedical literature: text mining of abstract vs. text mining of full text articles. *Knowledge Exploration in Life Science Informatics*, 3303, pp. 96-108.

Mayfield, J., McNamee, P. & Piatko, C. (2003). Named entity recognition using hundreds of thousands of features. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*. Edmonton, Canada, pp. 184-187.

Maynard, D., Tablan, V., Ursu, C., Cunningham, H. & Wilks, Y. (2001). Named entity recognition from diverse text types. In *Proceedings of Recent Advances in Natural Language Processing Conference*. CiteSeer, pp. 257–274.

McCallum, A. & Jensen, D. (2003). A note on the unification of information extraction and data mining using conditional-probability, relational models. *Journal of Machine Learning Research*, pp. 79-86.

McCallum, A. & Li, W. (2003). Early results for named entity recognition with conditional random fields. In *Proceedings of the 7th Conference on Natural Language Learning (CoNLL-2003)*, pp. 188-191.

McCallum, A., Freitag, D. & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*. Palo Alto, CA, pp. 591-598.

McFarlane, A. (2002). Educating the inheritors of information age. *Inaugural lecture, University of Bristol*, 18th November.

McFarlane, A. (2001). Perspectives on the relationships between ICT and assessment. *Journal of Computer Assisted Learning*, 17(3), pp. 227-234.

McIntyre, N. & Lapata, M. (2009). Learning to tell tales: A data-driven approach to story generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Singapore, pp. 217-225.

McKeachie, W.J. (2002). *McKeachie's Teaching Tips: Strategies, Research and Theory for College and University Teachers* 11th ed., Boston: Houghton Mifflin.

Mehdi, Y.-M., Farzindar, A. & Lapalme, G. (2010). Supervised machine learning for summarizing legal documents. In *Proceedings of the Canadian Conference on Artificial Intelligence*. Ottawa, Canada: Springer Berlin / Heidelberg, pp. 51-62.

Meulder, F.D. & Daelemans, W. (2003). Memory-based named entity recognition using unannotated data. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*. Edmonton, Canada, pp. 208-211.

Mikheev, A. (1999). A knowledge-free method for capitalized word disambiguation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 159-166.

Mitkov, R., Ha, L.A., Varga, A. & Rello, L. (2009). Semantic similarity of distractors in multiple-choice tests: Extrinsic evaluation. In *Proceedings of EACL 2009*

Workshop on GEometrical Models of Natural Language Semantics (GEMS-2009). Athens, Greece: Association for Computational Linguistics, pp. 49-56.

Mitkov, R., Ha, L.A. & Karamanis, N. (2006). A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(2), pp. 177-194.

Mitkov, R. & Ha, L.A. (2003). Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications using Natural Language Processing*, Edmonton, Canada, pp. 17-22.

Mohammad, S. & Hirst, G. (2005). Distributional measures as proxies for semantic relatedness. *Kluwer Academic Publishers, Netherlands*, pp. 1-48.

Mooney, R.J. & Bunescu, R. (2005). Mining knowledge from text using information extraction. *ACM SIGKDD Explorations Newsletter*, 7(1), pp. 3-10.

Mostow, J. & Chen, W. (2009). Generating instruction automatically for the reading strategy of self-questioning. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED).* Brighton, UK, pp. 465-472.

Mukherjea, S. & Sahay, S. (2006). Discovering biomedical relations utilizing the World-wide Web. *Pacific Symposium on Biocomputing*, 175(1), pp. 164-175.

Nadeau, D. & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1), pp. 3-26.

Nadeau, D., Turney, P.D. & Matwin, S. (2006). Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Proceedings of the 19th Canadian Conference on Artificial Intelligence*. Quebec, Canada, pp. 266-277.

Nielsen, R. (2008). Question generation: Proposed challenge tasks and their evaluation. In *Proceedings of the 1st Workshop on the Question Generation Shared Task and Evaluation Challenge*. Arlington, VA.

Ono, T., Hishigaki, H., Tanigami, A. & Takagi, T. (2001). Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics (Oxford, England)*, 17(2), pp. 155-61.

Palmer, D.D. & Day, D.S. (1997). A statistical profile of the named entity task. In *Proceedings of the 5th Conference on Applied Natural Language Processing*. Association for Computational Linguistics, pp. 190-193.

Palmer, M., Gildea, D. & Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1), pp. 71-106.

Palmer, M. & Finin, T. (1990). Workshop on the evaluation of natural language processing systems. *Computational Linguistics*, 16(3), pp. 175-181.

Papasalouros, A., Kotis, K. & Kanaris, K. (2008). Automatic generation of multiple-choice questions from domain ontologies. In *Proceeding of the IADIS e-Learning*. Amsterdam: IADIS, pp. 427-434.

Paroubek, P., Chaudiron, S. & Hirschman, L. (2007). Principles of evaluation in Natural Language Processing. *Traitement Automatique des Langues*, 48(1), pp. 7–31.

Paşca, M., Lin, D., Bigham, J., Lifchits, A. & Jain, A. (2006). Names and similarities on the web: Fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. pp. 809-816.

Paşca, M., Lin, D., Bigham, J., Lifchits, A. & Jain, A. (2006). Organizing and searching the World Wide Web of facts - Step one: The que-million fact extraction challenge. In *Proceedings of the National Conference on Artificial Intelligence*. pp. 1400-1405.

Pekar, V., Krkoska, M. & Staab, S. (2004). Feature weighting for co-occurrence-based classification of words. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04).* Geneva, Switzerland, pp. 799-805.

Pereira, F., Tishby, N. & Lee, L. (1993). Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL-93).* Columbus, Ohio, pp. 183-190.

Petasis, G., Vichot, F., Wolinski, F., Paliouras, G., Karkaletsis, V. & Spyropoulos, D. (2001). Using machine learning to maintain rule-based named-entity recognition and classification systems. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 426-433.

Pino, J. & Eskenazi, M. (2009). Semi-automatic generation of cloze question distractors effect of students' L1. In *Proceedings of the SLaTE Workshop on Speech and Language Technology in Education*. pp. 1-4.

Pino, J., Heilman, M.J. & Eskenazi, M. (2008). A selection strategy to improve cloze question quality. In *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains, 9th International Conference on Intelligent Tutoring Systems.*

Plake, C., Schiemann, T., Pankalla, M., Hakenberg, J. & Leser, U. (2006). AliBaba: PubMed as a graph. *Bioinformatics*, 22(19), pp. 2444-2445.

Pollock, M., Whittington, C. & Doughty, G. (2000). Evaluating the costs and benefits of changing to CAA. In *Proceedings of the 4th Annual CAA Conference*. Loughborough.

Ponomareva, N., Gomez, J.M. & Pekar, V. (2009). AIR: a Semi-Automatic System for Archiving Institutional Repositories. In *Proceedings of 14th International Conference on Applications of Natural Language to Information Systems (NLDB-09).* Saarbrucken, Germany: Springer Berlin / Heidelberg, pp. 169-181.

Popescu, A.-M. & Etzioni, O. (2005). Extracting product features and opinions from reviews. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-05).* Association for Computational Linguistics, pp. 339-346.

Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J.H. and Jurafsky,D. (2005). Support vector learning for semantic argument classification. *Machine Learning*, 60(1-3), pp. 11-39.

Pulman, S.G. & Sukkarieh, J.Z. (2005). Automatic short answer marking. In *Proceedings of 2nd Workshop on Building Educational Applications using NLP, Association for Computational Linguistics*, Ann Arbor, Michigan, pp. 9-16.

Pustejovsky, J., Castano, J., Zhang, J., Kotecki, M. & Cohran, B. (2002). Robust relational parsing over biomedical literature: Extracting inhibit relations. In *Proceedings of the 7th Annual Pacific Symposium on Biocomputing*. pp. 362-373.

Pustejovsky, J., Castano, J., Cohran, B., Kotecki, M. & Morrell, M. (2001). Automatic extraction of acronym-meaning pairs from MEDLINE databases. *Studies in Health Technology and Informatics*, 84(Pt 1), pp. 371-375.

Quinlan, J.R. (1986). Induction of decision trees. J. W. Shavlik & T. G. Dietterich, eds. *Machine Learning*, 1(1), pp. 81-106.

Rao, C.R. (1982). Diversity: Its measurement, decomposition, apportionment and analysis. *Sankhyà: The Indian Journal of Statistics Series A*, 44, Series (1), pp.1–22.

Ravichandran, D. & Hovy, E. (2002). Learning surface text patterns for a Question Answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL-02)*. Philadelphia, PA.

Reiter, E., Sripada, S., Hunter, J., Yu, J. & Davy, I. (2005). Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2), pp. 137-169.

Reiter, E. & Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press.

Riloff, E. & Jones, R. (1999). Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99)*. pp. 474-479.

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence*. AAAI Press, pp. 1044-1049.

Riloff, E. (1993). Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-93)*. AAAI Press/The MIT Press, pp. 811-816.

Sætre, R., Sagae, K. & Tsujii, J. (2007). Syntactic features for protein-protein interaction extraction. In *Proceedings of the 2nd International Symposium on Languages in Biology and Medicine (LBM-2007)*. Singapore, pp. 6.1-6.14.

Salton, G. & McGill, M.J. (1983). *Introduction to Modern Information Retrieval*, McGraw-Hill.

Sarawagi, S. (2008). Information extraction. *Foundations and Tends*, 1(3), pp. 261-377.

Sarmento, L., Jijkuon, V., de Rijke, M. & Oliveira, E. (2007). “More like these”: Growing entity classes from seeds. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM)*. Lisbon, Portugal, pp. 959-962.

Savova, G.K. Kipper-Schuler, K.C., Buntrock, J.D. & Chute, C.G. (2008). UIMA-based clinical information extraction system. In *Proceedings of LREC 2008 Workshop: Towards Enhanced Interoperability for Large HLT Systems UIMA for NLP*.

Scheuermann, F. & Guimarães Pereira, A. (2008). Towards a Research Agenda on Computer-based Assessment Challenges and needs for European Educational Measurement.

Schwartz, L., Aikawa, T. & Pahud, M. (2004). Dynamic language learning tools. In *Proceedings of the InSTIL/ICALL Symposium.*

Sekine, S. (2006). On-demand information extraction. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*. Association for Computational Linguistics, pp. 731-738.

Sekine, S. & Nobata, C. (2004). Definition, dictionary and tagger for extended named entities. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*. Lisbon, Portugal.

Sekine, S. (1998). NYU: Description of the Japanese NE System used for MET-2. In *Proceedings of the Seventh Message Understanding Conference (MUC-7).* Virginia, USA.

Shinyama, Y. & Sekine, S. (2006). Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics.* pp. 304-311.

Shinyama, Y. & Sekine, S. (2004). Named entity discovery using comparable news articles. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04).* Association for Computational Linguistics, pp. 848-853.

Siegfried, J.J. & Kennedy, P.E. (1995). Does pedagogy vary with class size in introductory economics? *American Economic Review, American Economic Association*, 85(2), pp. 347-351.

Skalban, Y. (2009). *Improving the output of a multiple-choice test generator: Analysis and proposals*. University of Wolverhampton.

Smith, S., Kilgarriff, A., Sommers, S., Wen-liang, G. & Guang-zhong, W. (2009). Automatic cloze generation for English proficiency testing. In *Proceedings of the LTTC Conference*. Taipei, Taiwan.

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1), pp. 233-272.

Soderland, S., Fisher, D. & Aseltine, J. (1995). CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*. pp. 1314-1319.

Soderland, S. & Lehnert, W. (1994). Corpus-driven knowledge acquisition for discourse analysis. In *Proceedings of AAAI'1994*. pp. 827-832.

Sparck, J.K. & Galliers, J. (1996). *Evaluating natural language processing systems: An analysis and review*. Lecture Notes in Artificial Intelligence, 1083, Berlin, Germany: Springer Verlag.

Stevenson, M. & Greenwood, M. (2009). Dependency pattern models for information extraction. *Research on Language and Computation*, 7(1), pp. 13-39.

Stevenson, M. & Greenwood, M. (2006). Comparing information extraction pattern models. In *Proceedings of the Information Extraction Beyond the Document Workshop (COLING/ACL 2006)*. Sydney, Australia, pp. 12-19.

Stevenson, M. & Greenwood, M. (2005). A semantic approach to IE pattern induction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL-05)*. Ann Arbor, Michigan, pp. 379-386.

Stevenson, M. & Ciravegna, F. (2003). Information extraction as a semantic web technology: Requirements and promises. In *Proceedings of the 14th European Conference on Machine Learning (ECML 2003) Workshop: Adaptive Text Extraction and Mining (ATEM-03)*. Cavtat-Dubrovnik, Croatia.

Stiggins, R.J. (2001). The unfulfilled promise of classroom assessment. *Educational Measurement Issues and Practice*, 20(3), pp. 5-15.

Sudo, K., Sekine, S. & Grishman, R. (2003). An improved extraction pattern representation model for automatic IE pattern acquisition. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, pp. 224-231.

Sudo, K., Sekine, S. & Grishman, R. (2001). Automatic pattern acquisition for Japanese information extraction. In *Proceedings of the 1st International Conference on Human Language Technology Research*. Association for Computational Linguistics, pp. 1–7.

Sumita, E., Sugaya, F. & Yamamoto, S. (2005). Measuring non-native speakers' proficiency of English by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the 2nd Workshop on Building Educational Applications using NLP*, June, pp. 61-68.

Szpektor, I., Tanev, H., Dagan, I. & Coppola, B. (2004). Scaling web-based acquisition of entailment relations. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*. Barcelona, Spain, pp. 41–48.

Tapanainen, P. & Järvinen, T. (1997). A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*. Washington, DC: Association for Computational Linguistics, pp. 64-74.

Tateno, J., Sano, H., Aizawa, H., Nakamura, T. & Morita, Y. (2005). Producing English educational materials from the BNC and releasing them on the Web. *IEIC Technical Report (Institute of Electronics, Information and Communication Engineers)*, 105(437), pp. 7-12.

Tjong Kim Sang, E. & Meulder, F.D. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the CoNLL-2003*. Edmonton, Canada, pp. 142-147.

Tjong Kim Sang, E. (2002). Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of the CoNLL-2002*. Taipei, Taiwan, pp. 155-158.

Tsuruoka, Y., Tateishi, Y., Kim, J-D., Ohta, T., McNaught, J., Ananiadou, S. & Tsujii, J. (2005). Developing a robust part-of-speech tagger for biomedical text. *Advances in Informatics – 10th Panhellenic Conference on Informatics, LNCS*, 3746, pp. 382-392.

Tsuruoka, Y. & Tsujii, J. (2005). Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-05).* Association for Computational Linguistics, pp. 467-474.

Turney, P.D. & Littman, M. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems TOIS*, 21(4), pp. 315-346.

Turney, P.D. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning (ECML-2001).* Freiburg, Germany, pp. 491-502.

Ureel, L., Forbus, K., Riesbeck, C. & Birnbaum, L. (2005). Question generation for learning by reading. In *Proceedings of the AAAI Workshop on Textual Question Answering*. Pittsburgh, Pennsylvania.

Vanderwende, L. (2008). The importance of being important: Question generation. In *Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge*. Arlington, VA.

Vanderwende, L. (2007). Answering and questioning for machine reading. In *Proceedings of the 2007 AAAI Spring Symposium on Machine Reading*. Stanford, CA.

Walker, M.A., Rambow, O. & Rogati, M. (2001). SPoT: A trainable sentence planner. In *Proceedings of the 2nd Annual Meeting of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 1-8.

Wang, X., Mohanty, N. & McCallum, A. (2005). Group and topic discovery from relations and text. In *Proceedings of the 3rd International Workshop on Link Discovery (LinkKDD-05)*. ACM Press, pp. 28-35.

Weeds, J. (2003). *Measures and applications of lexical distributional similarity*. PhD thesis, University of Sussex.

Weller, M. (2002). Assessment Issues on a web-based course. *Assessment & Evaluation in Higher Education*, 27(2), pp. 109-116.

Wilbur, J. & Smith, L. (2007). Biocreative 2. gene mention task. In *Proceedings of the 2nd Biocreative Challenge Evaluation Workshop*. pp. 7–16.

Wong, L. (2001). PIES, a Protein Interaction Extraction System. *Pacific Symposium on Biocomputing*, 531, pp. 520-531.

Yangarber, R. (2003). Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL-03)*. pp. 343-350.

Yangarber, R. (2000). *Scenario customization of information extraction*. PhD thesis, New York University.

Yangarber, R. & Grishman, R. (2000). Machine learning of extraction patterns from unannotated corpora: position statement. In *Proceedings of 14th European Conference* *on Artificial Intelligence (ECAI 2000), Workshop on Machine Learning for Information Extraction*. Berlin, Germany, pp. 76-83.

Yangarber, R., Grishman, R. & Tapananien, P. (2000). Unsupervised discovery of scenario-level patterns for information extraction. In *Proceedings of the 6th Conference on Applied Natural Language Processing*. Association for Computational Linguistics, pp. 282–289.

Yeh, A., Morgan, A., Colosimo, M. & Hirschman, L. (2005). BioCreAtIvE Task 1A: Gene mention finding evaluation. *BMC Bioinformatics*, 6(Suppl 1), S2.

Yuret, D. & Yatbaz, M.A. (2010). The noisy channel model for unsupervised word sense disambiguation. *Computational Linguistics*, 36(1), pp. 111-127.

Zelenko, D., Aone, C. & Richardella, A. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3(6), pp. 1083-1106.