The Sixth PASCAL Recognizing Textual Entailment Challenge

Luisa Bentivogli[1], Peter Clark,[2] Ido Dagan[3], Danilo Giampiccolo[4]

[1]FBK-irst
Trento, Italy
bentivo@fbk.eu

[2]Vulcan Inc.
Seattle, WA, USA
peterc@vulcan.com

[3]Bar-Ilan University
Ramat Gan, Israel
dagan@cs.biu.ac.il

[4]CELCT
Trento, Italy
giampiccolo@celct.it

Abstract

This paper presents the Sixth Recognizing Textual Entailment (RTE-6) challenge. This year a major innovation was introduced, as the traditional Main Task was replaced by a new task, similar to the RTE-5 Search Pilot, in which Textual Entailment is performed on a real corpus in the Update Summarization scenario. A subtask was also proposed, aimed at detecting novel information. To continue the effort of testing RTE in NLP applications, a KBP Validation Pilot Task was set up, in which RTE systems had to validate the output of systems participating in the KBP Slot Filling Task. Eighteen teams participated in the Main Task (48 submitted runs) and 9 in the Novelty Detection Subtask (22 submitted runs). As for the Pilot, 10 runs were submitted by 3 participants. Finally, the exploratory effort started in RTE-5 to perform resource evaluation through ablation tests was not only reiterated in RTE-6, but also extended to tools.

1 Introduction

The Recognizing Textual Entailment (RTE) task consists of developing a system that, given two text fragments, can determine whether the meaning of one text is entailed, i.e. can be inferred, from the other text. Since it inception in 2005, RTE has enjoyed a constantly growing popularity in the NLP community, as it seems to work as a common framework in which to analyze, compare and evaluate different techniques used in NLP applications to deal with semantic inference, a common issue shared by many NLP applications. After the first three highly successful PASCAL RTE Challenges held in Europe, RTE became a track at the Text Analysis Conference (TAC 2008), bringing it together with communities working on NLP applications. The interaction has provided the opportunity to apply RTE systems to specific application settings and move them towards more realistic scenarios. In particular, the RTE-5 Pilot Search Task represented a step forward, as for the first time Textual Entailment recognition was performed on a real text corpus. Furthermore, it was set up in the Summarization setting, attempting to analyze the potential impact of Textual Entailment on a real NLP application.

In an effort to catch the momentum and capitalize on the promise of the RTE-5 Pilot Search Task and the positive reception by the partici-

pants, the sixth round of the RTE challenges had two major objectives. First, it aimed to continue the research in RTE and sustain the advance of the state of the art in the field, by proposing data sets which reflect the natural distribution of entailment in a corpus and present all the typical problems that a system may deal with while detecting Textual Entailment in a natural setting, such as the interpretation of sentences in their discourse context. Second, RTE-6 aims to further explore the contribution that RTE engines can provide to Summarization applications.

In order to achieve these goals, major innovations were introduced in RTE-6. For the first time in 2010, the traditional Main Task which was carried out in the first five RTE challenges was not offered. Unlike the traditional Main Task framework, in which the data sets were composed of isolated, artificially created T(ext) – H(ypothesis) pairs, the new RTE-6 Main Task consists of recognizing Textual Entailment within a corpus. The task is situated in the Summarization setting and is a close variant of the RTE-5 Pilot Task: given a corpus, a hypothesis H, and a set of "candidate" sentences retrieved by the Lucene search engine from that corpus for H, RTE systems are required to identify all the sentences that entail the H among the candidate sentences.

In addition to the Main Task, a Novelty Detection Subtask was also proposed in RTE-6. The Novelty Detection Subtask is based on the Main Task and consists of judging whether the information contained in each hypothesis H is novel with respect to - i.e. not entailed by - the information contained in the corpus. The Novelty Detection Task is aimed at specifically addressing the needs of the Summarization Update scenario, where Summarization systems are required to write a short summary of a set of newswire articles, under the assumption that the user has already read a given set of earlier articles, and thus is interested in novel information.

Another major innovation introduced this year is represented by the new RTE-6 Pilot Task. In fact, the successful experience of the Pilot Task offered within RTE-5 pointed out opportunities to broaden the interaction between RTE and other application areas at TAC. To this purpose, a Knowledge Base Population (KBP) Validation Task was proposed as a Pilot within RTE-6. This task was based on the TAC KBP Slot Filling Task (McNamee and Dang, 2009), and was meant to show the potential utility of RTE systems for Knowledge Base Population.

Finally, following the positive experience of the fifth challenge, all the participants in the RTE-6 Main Task were asked to carry out ablation tests on the knowledge resources used by their systems, with the aim of studying the relevance of such resources in recognizing Textual Entailment. This year ablation tests were also extended to tools, such as parsers, coreference resolvers, Named Entity recognizers.

This paper describes the preparation of the data sets for both Main and Pilot tasks, the metrics used for the evaluation of the systems' submissions, and a preliminary analysis of the results of the challenge. In Section 2 the new Main Task is presented, describing the data sets, the evaluation methodology, and an analysis of the results achieved by the participating systems. Section 3 is dedicated to a detailed presentation of the Novelty Detection Subtask, and the KBP Validation Pilot Task is described in Section 4. In Section 5 the RTE-6 ablation tests, together with the RTE Knowledge Resources initiative, are presented. Conclusions and perspectives on future work are outlined in Section 6.

2 The RTE-6 Main Task: Recognizing Textual Entailment within a Corpus

Textual Entailment is defined as a directional relationship between two text fragments - T, the entailing text and H, the entailed text - so that *T entails H if, typically, a human reading T would infer that H is most likely true* (Dagan et al., 2006).

This definition of entailment is based on (and assumes) common human understanding of language as well as background knowledge; in fact, for Textual Entailment to hold it is required that *text and knowledge entail H, **but** knowledge alone cannot entail H.* This means that H may be entailed by incorporating some prior knowledge that would enable its inference from T, but it should not be entailed by that knowledge alone. In other words, H is not entailed if H is true regardless of T.

The traditional RTE Main Task, which was carried out in the first five RTE challenges, con-

sisted of making entailment judgments over isolated T-H pairs. In such a framework, both Text and Hypothesis were artificially created in a way that they did not contain any references to information outside the T-H pair. As a consequence, the context necessary to judge the entailment relation was given by T, and only language and world knowledge were needed, while knowledge of the textual context surrounding T was not required.

In contrast, the task of Recognizing Textual Entailment within a corpus, introduced as a pilot task in RTE-5 (see Bentivogli et al., 2009b), consists of finding all the sentences in a set of documents that entail a given Hypothesis. In such a scenario, both T and H are to be interpreted in the context of the corpus, as they rely on explicit and implicit references to entities, events, dates, places, situations, etc. pertaining to the topic[1].

In RTE-6, the traditional Main Task is replaced by the task of Recognizing Textual Entailment within a corpus.

The RTE-6 Main Task is situated in the Summarization application setting. To apply RTE in this setting (i) the RTE corpus is taken from the 2009 Summarization Task data set and (ii) the Hs are (standalone versions of) sentences in that data set, selected from those incorporated into the automatic summaries created by the systems participating in the Update Summarization Task[2].

The goal of the task is to further explore the contribution that RTE engines can make to Summarization. In fact, in a general summarization setting, correctly extracting all the sentences entailing a given candidate statement for the summary (similar to Hs in RTE) corresponds to identifying all its mentions in the text, which is useful for assessing the importance of that candidate statement for the summary and, at the same time, detecting those sentences which contain redundant information and should probably not be included in the summary.

[1] For an analysis of the relevance of discourse phenomena in Textual Entailment see (Bentivogli et al., 2009a).

[2] In the 2009 Summarization Task, the automatic summaries were an assembly of (sometimes modified) selected corpus sentences rather than synthesized sentences.

The rest of Section 2 describes the Main Task in detail, presenting a description of the task, the resulting data set, the metrics used to evaluate the systems' submissions and the results obtained.

2.1 Task Description

In the RTE-6 Main Task, given a corpus, a hypothesis H, and a set of "candidate" entailing sentences for that H retrieved by Lucene from the corpus, RTE systems are required to identify all the sentences that entail H among the candidate sentences.

Although the task is a close variant of the RTE-5 Pilot Task; it differs significantly in two ways. First, unlike in RTE-5, where the Search Task was performed on the whole corpus, in RTE-6 the search is just over the Lucene-retrieved candidates, and thus a preliminary Information Retrieval filtering phase was performed by the organizers while building the data set, as might be expected in any RTE system working with large amounts of text.

For this filtering phase, the retrieval component has to consider (i) each hypothesis as a query and (ii) the corpus sentences as "the documents" to be retrieved. To this purpose, the Apache Lucene [3] text search engine, Version 2.9.1, was used with the following characteristics:

- *StandardAnalyzer* (tokenization, lowercase and stop-word filtering, basic clean-up of words)
- Boolean "OR" query
- Default document scoring function

In order to decide which Lucene setting was best, an experiment was conducted on the whole RTE-5 Search data set and on three topics of the RTE-6 Development Set. Results showed that, when the first 100 top-ranked sentences for each H are taken as candidates, Lucene achieves a recall of about 0,80. This appeared to be a good compromise, as it provided a sufficient number of entailing sentences, while also being a manageable number to create a gold standard annotation. However, it is worth noting that this choice implied that about 20% of entailing sentences,

[3] http://lucene.apache.org/

present in the corpus but not retrieved by Lucene, got lost in this RTE-6 exercise.

A second important difference compared with the RTE-5 Search Task is the fact that a certain number of Hs have no entailing sentences in the corpus, and also that some documents in the corpus do not contain any entailing sentences.

The example below presents a hypothesis (H) referring to a given topic, and some of the entailing sentences (T) among the larger set of candidate sentences retrieved by Lucene:

H Jill Carroll was seized by gunmen

T The Christian Science Monitor newspaper on Monday pleaded for the release of American reporter Jill Carroll, seized in Baghdad by abductors who gunned down her Iraqi translator. *(*doc_id="AFP_ENG_20060109.0574" s_id="1"*)*

T US-Iraq-journalist-kidnap WASHINGTON: The Christian Science Monitor newspaper pleaded for the release of American reporter Jill Carroll, seized in Baghdad by abductors who gunned down her Iraqi translator. *(*doc_id="AFP_ENG_20060110.0001" s_id="9"*)*

T Jill Carroll, 28, a freelance reporter working for the Christian Science Monitor, was seized by gunmen on Saturday after calling by the office of a prominent Sunni politician, the US newspaper confirmed on Monday. *(*doc_id="AFP_ENG_20060110.0024" s_id="2"*)*

T The 28-year-old reporter was seized by gunmen on Saturday after calling by the office of a prominent Sunni politician in the neighbourhood. *(*doc_id="AFP_ENG_20060110.0430" s_id="7"*)*

It is important to note that while only the subset of the candidate entailing sentences must be judged for entailment, these sentences are not to be considered as isolated texts. Rather, the entire corpus to which the candidate entailing sentences belong is to be taken into consideration in order to resolve discourse references and appropriately judge the entailment relation. For instance, the last sentence (s_id="7") in the example above was considered as entailing the H because, from its context, it could be understood that the mention "The 28 year-old reporter" refers to the entity "Jill Carroll", mentioned earlier in the discourse.

2.2 Data Set Description

The RTE-6 Main data set is based on the data created for the TAC 2009 Update Summarization Task. The TAC 2009 SUM Update data consists of a number of topics, each containing two sets of documents, namely (i) Cluster A, made up of the first 10 texts in chronological order (of publication date), and (ii) Cluster B, made up of the last 10 texts.

The RTE-6 data set is composed of 20 topics, 10 used for the Development Set and 10 for the Test Set. For each topic, the RTE-6 Main Task data consist of:

a) Up to 30 Hypotheses referring to the topic. The Hs are standalone versions of sentences in the Cluster B documents.
b) A set of 10 documents, corresponding to the Cluster A corpus.
c) For each H, a list of up to 100 candidate entailing sentences (the Ts) from the Cluster A corpus and their location in the corpus.

While Ts are naturally occurring sentences in a corpus and are to be taken as they are, the Hs were slightly modified from the originals so as to be standalone sentences. The procedure applied for the creation of the Hs is described in the following section.

2.3 Creation of the Hypotheses

In order to be as consistent as possible with the Summarization scenario, the Hs were standalone versions of the Cluster B sentences included in the automatic summaries of Cluster B documents.

Our original goal was that all the content of the automatic summaries was captured by the Hs. To do that, first all the sentences present in the 10 best scoring systems[4] participating in the TAC 2009 Update Summarization Task were collected. When a summary sentence contained several pieces of information, it was divided into simpler content units, which were then rephrased as standalone sentences. For example, from the summary sentence "*Merck, the maker of Vioxx, which was approved by the FDA in 1999, voluntarily took the drug off the market in September.*", taken from Topic 924 in Development Set, three different Hs were created, namely H147 "*Merck is the maker of Vioxx.*"; H151"*Vioxx was approved by the FDA in*

[4] According to the Pyramid evaluation results for summaries of Cluster B (see Dang and Owczarzak, 2009).

1999."; and H153 "*Merck withdrew Vioxx from the market.*"

As can be seen in the example above, although we strove to preserve the original sentences as verbatim as possible, minor syntactic and morpho-syntactic changes were allowed, if needed to produce grammatically correct standalone sentences and resolve all the discourse references. Although our goal was to capture all the content of the automatic summaries, in the end not all the content was represented, due to practical constraints. For example, if the number of the Hs needed to represent all the information contained in the summaries exceeded the maximum of 30, some Hs were discarded[5]. Also, in the example above, the information "in September" contained in the summary was not included in H153, in order to maximize the number of entailing sentences in Corpus A. However, it is fair to say that the information present in the automatic summaries was largely represented by the Hs.

In addition, in order to obtain a sufficient number of entailing sentences necessary for the RTE task, an additional number of Hs was created directly from the Cluster B corpus text snippets, even if not present in the automatic summaries. Thus the overall set of Hs (for the Main Task, but not the for the Novelty Subtask), went beyond the information in the summaries.

As regards T and H time anchoring, the time of H is always later than the time of T, due to the fact that Hs are taken from summaries of Cluster B, which is made up of more recent documents.

For T and H verb tenses, since verb tenses are intrinsically deictic and depend on their anchor time, systems must take into account that both Ts and Hs are naturally anchored to the publication date of the document from which they are taken (for more detail, see Bentivogli et al., 2009a).

2.4 The Final Data Set

The Development Set is composed of 10 topics, and contains globally 221 Hs, 28 of which were not taken from the automatic summaries but directly from cluster B sentences. For each H of a topic, all the candidate entailing sentences (100 at most) had to be judged for entailment, yielding 15,955 sentence annotations, of which 897 are "entailment" judgments (note that the same sentence can be a candidate for - and entail - more than one H). 89 Hs do not have entailing sentences, while the remaining 122 have at least one entailing sentence.

The Test Set is also composed of 10 topics, and contains globally 243 Hs, 44 of which were not taken from the automatic summaries but directly from cluster B sentences. There are 19,972 sentence annotations, 945 of which are "entailment" judgments. 100 Hs do not have entailing sentences, while the remaining 143 have at least one entailing sentence.

In order to assure the creation of a high quality resource, the whole data set was annotated by three assessors. Once the annotation was performed, a reconciliation phase was carried out to eliminate annotators' mistakes and leave only real disagreements. After the reconciliation phase, the inter-annotator agreement calculated using the Kappa statistics (Siegel and Castellan, 1988; Fleiss, 1971) was 98.36% for the Development Set and 97.83% for the Test Set[6].

2.5 Evaluation Measures

The evaluation was carried out in the same way as in the RTE-5 Search Task. System results were compared to a human-annotated gold standard and the metrics used to evaluate system performances were Precision, Recall, and F-measure.

The official metric chosen for ranking systems was micro-averaged F-measure. Additionally, macro-averaged results for topics were made available to participants. As systems were not forced to retrieve at least one entailing sentence for each topic, in order to calculate macro-averaged results it was decided that, if no sentence was returned for a given topic, the Precision for that topic is 0. Moreover, as a high number of Hs had no entailing sentences, macro-

[5] Some criteria were followed in selecting the Hs, such as choosing i) all the Hs that had entailing sentences in the corpus, and ii) among those that had no entailing sentences, only the Hs generated from the text snippets which were present in most summaries.

[6] It is worth mentioning that the percentage of agreement over those annotations where at least one assessor said YES was 95.42% for the Development Set and 94.34% for the Test Set.

RUN	Micro-Average			Macro-Average (by TOPIC)		
	Precision	**Recall**	**F-measure**	Precision	Recall	F-measure
BIU1	37.54	37.46	**37.5**	37.66	36.84	37.25
BIU2	29.4	36.08	32.4	31.05	34.77	32.81
BIU3	33.36	37.88	35.48	33.19	37.44	35.19
Boeing1	55.1	36.61	**43.99**	56.06	39.08	46.06
Boeing2	64.49	26.14	37.2	76.94	27.12	40.1
budapestacad1	13.35	31.22	**18.71**	18.12	31.85	23.1
budapestacad2	12.9	32.06	18.4	18.17	32.24	23.24
budapestacad3	12.01	12.38	12.19	13.49	11.96	12.68
deb_iitb1	55.98	34.18	42.44	58.25	33.96	42.91
deb_iitb2	53.43	42.86	**47.56**	55.75	42.9	48.49
deb_iitb3	71.61	30.16	42.44	73.99	30	42.69
DFKI1	53.31	29.84	38.26	60.19	29.24	39.36
DFKI2	55.94	30.9	**39.81**	61.85	30.2	40.58
DFKI3	56.08	27.83	37.2	62.1	28.07	38.67
DirRelCond21	38.99	41.8	**40.35**	41.34	42.22	41.77
DirRelCond22	52.38	15.13	23.48	54.23	15.6	24.24
DirRelCond23	61.76	17.78	27.61	63.76	18.89	29.15
FBK_irst1	35.09	49.21	40.97	37.24	50.43	42.84
FBK_irst2	33.36	49.95	40	35.56	51.19	41.97
FBK_irst3	43.46	46.03	**44.71**	45.95	46.77	46.35
IKOMA1	39.71	51.43	**44.81**	40.05	51.64	45.11
IKOMA2	39.59	51.53	44.78	39.86	51.43	44.91
IKOMA3	45.39	43.81	44.59	46.72	44.15	45.4
JU_CSE_TAC1	38.63	31.64	**34.79**	39.71	33.42	36.29
JU_CSE_TAC2	38.49	20.53	26.78	33.44	21.01	25.81
JU_CSE_TAC3	78.3	19.47	31.19	81.2	19.6	31.57
PKUTM1	70.14	36.3	47.84	72.75	37.15	49.18
PKUTM2	68.57	36.93	**48.01**	71.6	37.92	49.58
PKUTM3	68.69	35.98	47.22	71.46	36.88	48.65
Sagan1	15.98	48.89	**24.09**	16.54	50.88	24.97
Sagan2	14.31	50.37	22.29	14.99	51.74	23.24
Sagan3	13.39	41.06	20.19	13.95	43.12	21.08
saicnlp1	7.92	21.69	**11.6**	7.7	21.48	11.34
Sangyan1	21.66	46.03	**29.46**	21.92	46	29.69
SINAI1	23.4	24.76	24.06	25.66	26.4	26.02
SINAI2	23.27	30.69	**26.47**	25.07	32.07	28.14
SJTU_CIT1	26.34	57.78	36.18	26.8	57.27	36.51
SJTU_CIT2	32.09	49.95	39.07	32.04	49.22	38.81
SJTU_CIT3	34.35	46.67	**39.57**	34.13	45.79	39.11
UAIC20101	22.89	27.2	**24.85**	25.24	26.99	26.09
UAIC20102	14.02	39.15	20.64	15.46	38.27	22.03
UAIC20103	31.49	17.46	22.46	32.66	17.91	23.13
UB.dmirg1	12.22	13.44	12.8	12.58	13.42	12.99
UB.dmirg2	18.58	8.89	12.03	19.91	9.05	12.44
UB.dmirg3	11.79	48.68	**18.98**	12.48	49.79	19.96
UIUC1	46.25	23.49	31.16	53.86	24.74	33.91
UIUC2	38.11	26.46	31.23	40.34	27.82	32.93
UIUC3	31.53	33.86	**32.65**	38.86	36.43	37.6

Table 1. Main Task results (in bold Best run of each system)

averaged results for hypotheses were not calculated.

2.6 Submitted Systems and Results

Eighteen teams participated in the Search Task, submitting a total of 48 runs. Table 1 presents the micro- and macro-averaged results of all the submitted runs. Details about Precision, Recall, and F-measure for single topics can be found in the Notebook Appendix. As regards overall results on micro-average, Table 2 shows some F-measure statistics, calculated both (i) over all the submitted runs and (ii) considering only the best run of each participating group.

A first general analysis of the results shows that macro-averaged scores, although generally higher, are overall close to the micro-averaged ones. As far as a comparison of Precision and Recall is concerned, although Recall values are higher in more than half of participating systems, a quite large number performed better in Precision – sometimes significantly better, as in the case of the best performing system, which achieved a Precision of 68.57 compared to a Recall of 36.93. Considering the difference between Precision and Recall within each run, a large variability is noted between the systems, ranging (on micro-averaged results) from virtually no difference – 0.08 – for *BIU1* to 58.38 for *JU_CSE_TAC3*.

Five Information Retrieval baselines were also calculated. The results are shown in Table 3. The first four baselines were created considering as entailing sentences respectively the top 5, 10, 15, 20 sentences ranked by Lucene. The fifth baseline considered as entailing sentences all the candidate sentences to be judged for entailment in the Main Task, i.e. the top 100 sentences (at most) retrieved by Lucene. Table 3 shows that Baseline_5 performed best, scoring an F-measure of 34.63, which is below the median but above the average best results of participating systems.

F-measure	All runs	Best runs
Highest	48.01	48.01
Median	33.72	36.14
Average	32.30	33.77
Lowest	11.60	11.60

Table 2. Main Task F-measure statistics

	Precision	Recall	F1
Baseline_5	30.78	39.58	34.63
Baseline_10	21.87	56.19	31.49
Baseline_15	17.15	66.03	27.23
Baseline_20	14.23	72.70	23.80
Baseline_ALL	4.73	100.00	9.03

Table 3. Baseline results

Although a real comparison between the results of the RTE-5 Search is not possible, as the two exercises were similar but still different, it may be concluded that the outcome of the RTE-6 Main Task recorded a positive trend. In fact, a consistently larger number of participants tested their systems on the task of recognizing Textual Entailment within a corpus, and achieved an overall improvement of results. Moreover, while in RTE-5 the best result was below the baseline, this year the best baseline is below the median, and the best system's F-measure is 13.38 points above it.

Such promising results suggest that RTE techniques may be used, in addition to simple IR techniques, to help summarization systems in detecting sentences that imply each other, and thus removing duplicates.

3 Novelty Detection Subtask

The Novelty Detection Subtask is based on the Main Task, and uses a subset of the Main Task data. It consists of judging if the information contained in each H - drawn from the cluster B documents - is novel with respect to the information contained in the set of Cluster A candidate entailing sentences. If for a given H one or more entailing sentences are found, it means that the content of the H is not new. On the contrary, if no entailing sentences are detected, it means that the information contained in the H is novel.

The Novelty Detection Subtask was aimed at specifically addressing the needs of the Summarization Update Task. In this task, systems are required to write a short summary of a set of newswire articles, under the assumption that the user has already read a given set of earlier articles. In such a setting, it is important to distinguish between novel and non-novel information. RTE engines which are able to detect the novelty of Hs - i.e., find Hs which have no entailing Ts - can help Summarization systems filter out

non-novel sentences from their summaries. From the systems' point of view, the Novelty Detection Subtask was similar to the Main Task, and did not require any additional annotations. Rather, the novelty detection decision could be derived automatically from the number of entailing sentences found for each H: when no entailing sentences for that H were found among the Cluster A candidate entailing sentences, then the H was judged as novel. In contrast, if more than one entailing sentence was retrieved for a given H, then the H was judged as non-novel. As for the Main Task, for non-novel Hs all the entailing sentences had to be returned as justification of the judgment. Given this setting, the participants in the Subtask had the opportunity to tune their systems specifically for novelty detection, without having to change their output format.

Nevertheless, the Novelty Detection Task differed from the Main Task primarily because the Hs were only a subset of the Hs used for the Main Task, namely those taken from the automatic summaries. Moreover, the system outputs were scored differently, using specific scoring metrics designed for assessing novelty detection. In the following, both the data set and the evaluation metrics are described in detail.

3.1 The Data Set

The Novelty Detection data set was the same as the Main Task data set, except that it contained only the subset of the Hs taken from the automatic summaries and their corresponding candidate sentences. The Hs of the Main Task were taken both from automatic summaries and directly from the Cluster B documents. However, the Hs that were not taken from the automatic summaries, though necessary for the Textual Entailment task, were less interesting from a Summarization perspective. This was because they do not reflect the output of actual summarization systems, and they have relatively numerous entailing sentences in the Cluster A corpus and thus could be more easily recognized as non-novel by the summarization systems. For this reason, these Hs were excluded from the Novelty Detection data.

The resulting data set of the Novelty Detection Task is a subset of the Main Task data set. The Development Set is composed of 10 topics, and contains globally 183 Hs. Among them, 89 Hs contain novel information (i.e. they have no entailing sentences), whereas 94 Hs do not contain novel information, with a total number of entailing sentences of 707.

The Test Set is composed of 10 topics, and contains globally 199 Hs. Among them, 100 Hs contain novel information (i.e. they have no entailing sentences), whereas 99 Hs do not contain novel information, with a total number of entailing sentences of 723.

The inter-annotator agreement calculated using the Kappa statistics was 98.21% for the Development Set and 97.68% for the Test Set.

3.2 Evaluation Measures

As in the Main Task, the system results were compared to the human-annotated gold standard. Two scores were used to evaluate the system performance on the Novelty Detection Task, namely:

1) The primary score is Precision, Recall and F-measure computed on the binary novel/non-novel decision. The novelty detection decision was derived automatically from the number of justifications provided by the system - i.e. the entailing sentences retrieved for each H - where 0 implies 'novel', 1 or more 'non-novel'.

2) The secondary score measures the quality of the justifications provided for non-novel Hs, that is the set of all the sentences extracted as entailing the Hs. This type of evaluation is the same as the one carried out for the Main Task, and uses the same metrics, i.e. Micro-averaged Precision, Recall and F-measure.

3.3 Submitted Systems and Results

Nine teams participated in the Novelty Detection Task, submitting 22 runs.

Table 4 presents the results of the Novelty Detection and Justification scores for all the systems participating in the task. More details about Precision, Recall and F-measure for the single topics can be found in the Notebook Appendix.

For overall results on Novelty Detection and Justification scores, Table 5 shows some F-measure statistics, calculated both over all the submitted runs and considering only the best run of each participating group.

System performances were quite good in Novelty Detection. As the median of the best

RUN	Evaluation - Micro-Average			Justification - Micro-Average		
	Precision	Recall	F-measure	Precision	Recall	F-measure
BIU1	73.53	75	**74.26**	34.83	36.38	**35.59**
BIU2	71.43	70	70.71	26.94	34.58	30.28
BIU3	77.91	67	72.04	29.76	38.04	33.39
Boeing1	68.46	89	77.39	50.62	34.02	**40.69**
Boeing2	66.43	93	**77.5**	59.45	26.97	37.11
DFKI1	74.34	84	78.87	48.85	29.46	**36.76**
DFKI2	73.5	86	**79.26**	52.33	27.94	36.43
IKOMA1	82.02	73	77.25	39.91	49.52	44.2
IKOMA2	79.44	85	**82.13**	47.63	43.02	**45.2**
IKOMA3	75.65	87	80.93	53.32	35.55	42.66
JU_CSE_TAC1	80.58	83	**81.77**	40.92	29.6	**34.35**
JU_CSE_TAC2	71.67	86	78.18	41.26	19.92	26.87
JU_CSE_TAC3	66.67	96	78.69	75.71	14.66	24.57
PKUTM1	72.39	97	**82.91**	69.01	36.65	47.88
PKUTM2	72.73	96	82.76	67.75	37.48	**48.26**
PKUTM3	71.85	97	82.55	68.12	36.65	47.66
Sagan1	46.15	42	**43.98**	2.15	16.18	**3.79**
SINAI1	65.62	42	**51.22**	20.72	23.79	**22.15**
SINAI2	63.33	38	47.5	20.93	22.41	21.64
UAIC20101	81.4	70	75.27	21.91	27.94	**24.56**
UAIC20102	81.54	53	64.24	13.48	40.66	20.25
UAIC20103	73.28	85	**78.7**	30.22	17.43	22.11

Table 4. Novelty Detection task results (in bold the Best run of each system)

runs shows, more than half of the systems achieved an F-measure above 78, and three systems scored above 80. Also the average F-measure values were quite high, being 72.41 on the best runs. Recall was generally higher than Precision, reaching 97 in the best case (see Table 4). Nevertheless, the difference between Precision and Recall within each single run varied considerably, ranging from a minimum of 1.43 in *BIU2* to a maximum of 29.33 in *JU_CSE_TAC3.*

A baseline was calculated in which all the Hs are classified as novel. The baseline scored a Precision of 50.25, a Recall of 100, and a corresponding F-measure of 66.89. This baseline, which indicates the proportion of novel Hs in the Test Set, is below the average, and is outdone by the best run results of 7 out of 9 systems.

For Justification, the results aligned closely with the performances in the Main Task. Table 4 shows that a larger number of systems performed better on Precision than on Recall (in particular, *JU_CSE_TAC3*'s Precision is 61.05 points higher than Recall).

Overall these results seem promising, and suggest that Summarization systems could exploit the Textual Entailment techniques for novelty detection when deciding which sentences should be included in the Update summaries.

NOVELTY DETECTION			JUSTIFICATION (non novel Hs)	
F-measure	All runs	Best runs	All runs	Best runs
Highest	82.91	82.91	48.26	48.26
Median	77.84	78.70	34.97	35.59
Average	73.55	72.41	32.88	32.38
Lowest	43.98	43.98	3.79	3.79

Table 5. Novelty Detection F-measure statistics

4 Knowledge Base Population Validation Pilot Task

Continuing the effort to bring people from the three TAC communities together and to create a common framework in the field of text under-

standing, a new Knowledge Base Population (KBP) Validation Pilot was proposed, based on the TAC KBP Slot Filling Task (McNamee and Dang, 2009). The goal of this task was to show the potential utility of RTE systems for Knowledge Base Population, similar to the goals in the Summarization setting.

4.1 Task Description

The KBP Validation Pilot Task is situated in the Knowledge Base Population scenario and aims to validate the output of the systems participating in the KBP Slot Filling Task by using Textual Entailment techniques. The idea of using Textual Entailment to validate the output of NLP systems was partly inspired by a similar experiment, namely the Question Answering Task, performed as a part of the CLEF Campaign from 2006 to 2008 (Peñas et al., 2007).

The KBP Slot Filling Task, on which the Validation Task is based, consists of searching a collection of documents and extracting values for a pre-defined set of attributes ("slots") for target entities. In other words, given an entity in a knowledge base and an attribute for that entity, systems must find in a large corpus the correct value(s) for that attribute and return the extracted information together with a corpus document supporting it as a correct slot filler.

The RTE KBP Validation Pilot is based on the assumption that an extracted slot filler is correct if and only if the supporting document entails a hypothesis summarizing the slot filler. For example, consider the following slot filler and supporting document returned by a KBP system for the "residences" attribute for the target entity "Chris Simcox":

KBP System Input
- Target Entity: *"Chris Simcox"*
- Slot: Residences
- Document collection

KBP System Output
- Slot Filler: *"Tucson, Ariz."*
- Supp Doc: `NYT_ENG_20050919.0130.LDC2007T07`

If the slot filler is correct, then the document `NYT_ENG_20050919.0130.LDC2007T07` must entail one or more of the following Hypotheses, created from the slot filler:

H1: *Chris Simcox lives in Tucson, Ariz.*
H2: *Chris Simcox has residence in Tucson, Ariz.*
H3: *Tucson, Ariz. is the place of residence of Chris Simcox*
H4: *Chris Simcox resides in Tucson, Ariz.*
H5: *Chris Simcox's home is in Tucson, Ariz.*

In other words, the KBP Validation Task consists of determining whether a candidate slot filler is supported in the associated document using entailment techniques.

Each slot filler submitted by a system participating in the KBP Slot Filling Task results in one evaluation item (i.e. a T-H "pair") for the RTE-KBP Validation Pilot, where T is the source document that was cited as supporting the slot filler, and H is a set of simple, synonymous Hypotheses created from the slot filler.

A distinguishing feature of the KBP Validation Pilot is that the resulting T-H pairs differ from the traditional pairs because (i) T is an entire document, instead of a single sentence or a paragraph and (ii) H is not a single sentence but a set of roughly synonymous sentences representing different linguistic realizations of the same slot filler.

Another major characteristic of the KBP Validation Task, which distinguishes it from the other RTE challenges proposed so far, is that the RTE data set is created semi-automatically from KBP Slot Filling participants' submissions, and the gold standard annotations are automatically derived from the KBP assessments.

4.2 Data Set Description

The RTE-6 KBP Validation data set was based on the data created for the KBP 2009 and 2010 Slot Filling Task. More precisely, the Development Set was created from the 2009 KBP data, whereas the Test Set was created from KBP 2010 data.

The creation of the RTE-6 Pilot task data set was semi-automatic and took as starting points (i) the extracted slot-fillers from multiple systems participating in the KBP *Slot Filling* task and (ii) their assessments[7].

[7] As the Slot Filling task can be viewed as a more traditional Information Extraction task, the methodology used for creating the T-H pairs in this Pilot was the same as that adopted for the manual creation of IE pairs in the Main Task data sets from RTE-1 to RTE-5. In order to create those IE pairs, hypotheses were taken from the relations tested in the ACE tasks, while texts were extracted from the

During a first manual phase, before the automatic generation of the Hs for the data set, several "seed" linguistic realizations of templates were created for each target slot, expressing the relationship between the target entity and the extracted slot filler. For example, given the attribute "origin" belonging to a target entity of type "person", the following templates were manually created:

Template 1*:* ***X***'s origins are in ***Y***
Template 2*:* ***X*** comes from ***Y***
Template 3*:* ***X*** is from ***Y***
Template 4*:* ***X*** origins are ***Y***
Template 5*:* ***X*** has ***Y*** origins
Template 6*:* ***X*** is of ***Y*** origin

Then, each slot filler submitted by a system participating in the KBP Slot Filling Task became one evaluation item and was used to automatically create an RTE T-H pair. The T corresponded to the corpus document supporting the answer (as identified by the KBP system), while the H was created by instantiating all the templates for the given slot both with the name of the target entity (X) and the slot filler extracted by the system (Y). Providing all the instantiated templates of the corresponding slot for each system answer meant that each T-H pair does not contain only a single H, but rather a set of synonymous Hs. This setting has the property that for each example either all Hs for the slot are entailed or all of them are not.

The procedure adopted to create the Hs implied that some automatically generated Hs could be ungrammatical. While the Hs' templates are predefined, the slot fillers returned by the KBP systems are strings which can be incomplete, include extraneous text, or belong to a POS which is not compatible with that required by a specific H template. For instance, in the example below, given (i) the H templates for the slot "origin", (ii) the target person entity "Chris Simcox" and (iii) a correct slot filler "Canadian", both grammatical and ungrammatical Hs within the same evaluation item were obtained, i.e.:

H1: *Chris Simcox's origins are in Canadian*
H2: *Chris Simcox comes from Canadian*
H3: *Chris Simcox is from Canadian*
H4: *Chris Simcox origins are Canadian*
H5: *Chris Simcox has Canadian origin*
H6 *Chris Simcox is of Canadian origin*

These ungrammaticalities were left in the dataset.

The RTE gold standard annotations were automatically derived from the KBP assessments, converting them into Textual Entailment values. The assumption behind this process is that the KBP judgment of whether a given slot filler is correct coincides with the RTE judgment of whether the text entails the template instantiated with the target entity and the automatically extracted slot filler. As the KBP assessments were 4-valued, a mapping was necessary to convert KBP assessments into entailment values: "correct" and "redundant" KBP judgments were mapped into YES entailment; "wrong" judgments were mapped into NO entailment; and, as "inexact" judgments could result both in YES and NO entailment values, RTE pairs involving "inexact" KBP judgments were excluded from the data set.

As in all RTE data sets, temporal issues arise. However, as no temporal qualifications are defined for the KBP slots, differences in verb tense between the Hypothesis and Document Text in the RTE KBP Validation Task had to be ignored. For example, in the KBP Slot Filling Task, "*Tucson, Ariz.*" is considered a correct slot filler for the "residence" attribute of the target entity "Chris Simcox" if the supporting document contained the text "*Chris Simcox lived in Tucson, Ariz., before relocating to Phoenix*"; therefore, in the KBP Validation Task, the Hypothesis "*Chris Simcox lives in Tucson, Ariz.*" must be considered as entailed by the same document.

4.3 Final Data Set

The Development Set pairs were created from the 10,416 KBP 2009 Slot Filling Task assessments. After removing some pairs which were deemed unsuitable for the RTE-KBP Validation

outputs of actual IE systems, which were fed with relevant news articles. Correctly extracted instances were used to generate positive examples, and incorrect instances to generate negative examples.

Task[8], the final Development Set contained 9,462 T-H pairs, among which 694 pairs were positive examples (entailment value "YES"), and 8,768 were negative examples (entailment value "NO").

The KBP Validation Test Set was created from a different data collection, namely the KBP 2010 assessments. For this reason it differed from the Development Set in several ways. First, the size of the data set and the ratio between positive and negative pairs differed, as these depend on the number of KBP 2010 systems' submissions and on their performances respectively.

In addition, some changes were made to the KBP Slot Filling Task in 2010, which impacted the RTE Test Set. First, the proportion of Web documents with respect to newswire documents differed, as the 2010 KBP corpus contained a higher number of Web documents, which were generally longer. Second, some location slots, such as "*place of birth*", "*place of death*", "*residence*", and "*headquarters*" in the KBP 2009 task were expanded into separate, more specific slots in KBP 2010, such as "*city of birth*", "*state or province of birth*"; "*country of death*"; "*city of death", "state of death", etc.* Therefore, the KBP Validation H templates were changed accordingly, to reflect the more specific semantics of the slot.

The Test Set was created from the 24,014 KBP 2010 Slot Filling Task assessments. Once unsuitable pairs were removed[9], it contained 23,192 T-H pairs.

4.4 Evaluation Metrics

System results were compared to the gold standard created automatically from the KBP assessments of the systems' output. The system performances were measured calculating Micro-Averaged Precision, Recall, and F-measure.

RUN	GENERIC		
	Precision	**Recall**	**F-measure**
FBK_irst1	20.46	33.82	**25.5**
FBK_irst2	19.69	34.66	25.11
JU_CSE_TAC2	22.4	13.96	**17.2**
BIU2	10.02	39.48	**15.98**
BIU1	10.06	37.51	15.87
JU_CSE_TAC3	9.91	33.68	15.31
JU_CSE_TAC1	9.25	29.06	14.03
	TAILORED		
JU_CSE_TAC2	24.32	51.67	33.07
JU_CSE_TAC3	24.24	51.08	**32.88**
JU_CSE_TAC1	22.18	46.36	30

Table 6. KBP Validation results (in bold the Best run of each system)

4.5 Submitted Systems and Results

Two different types of submissions were allowed for this task:

- one for *generic* RTE systems, for which no manual effort was invested to tailor the generic system to the specific slots (beyond fully automatic training on the Development Set);
- the second for *manually tailored* systems, where it was allowed to invest additional manual effort to adapt the systems for the specific slots.

Three groups, all submitting runs for generic systems and one submitting tailored runs as well, participated in the KBP Validation Task, submitting a total of 10 runs – 7 generic and 3 tailored.

Table 6 presents the results, ranked according to F-measure scores. The median F-measure for the best generic runs is 17.2 (15.98 considering all runs), meanwhile the average value for best runs is 19.56 (18.43 considering all runs); on manually tailored submissions, the average value is 31.98. The manually tailored system performed significantly better than the generic systems. Recall was generally higher than Precision, both for generic and tailored systems, except in one case. More details about Precision,

[8] Different types of pairs were removed, namely (i) pairs involving GPE's – as we knew they wouldn't have been addressed in KBP2010; (ii) pairs for which the original KBP assessment was "inexact"; (iii) pairs involving KBP system answers of type "NO_RESPONSE"; (iv) duplicate KBP submissions (same answer and document); (v) pairs where the Ts were speech transcriptions, which were particularly difficult to process as did not contain punctuation and capitalization.

[9] Another type of pairs was not included in the Test Set, namely pairs involving "other_family" slots. The reason was that the templates created for this slot over-generated YES entailment judgments with respect to KBP "Correct" judgments.

Recall, and F-measure for each single Slot are given in the Notebook Appendix.

A baseline which classifies all Ts as entailing their corresponding Hs was calculated. The idea behind this baseline is that it reflects the cumulative performance of all KBP 2010 Slot Filling systems, as the RTE-KBP data set includes only Ts which were proposed as implying the corresponding H by at least one KBP system. The baseline, which also indicates the percentage of entailing pairs in the test set, scored a Precision of 8.77, a Recall of 100, and a corresponding F-measure of 16.13. Both median and average best results of RTE participating systems are above the baseline - with 2 out of 3 systems outperforming it, suggesting that a slot filling validation filter using RTE techniques could be useful for KBP systems.

The task proved to be particularly challenging for RTE systems, partly due to the difference between the Development and Test data. One of the common re marks from the participants was that it took longer than expected to process the Test Set, as it contained a high number of unexpectedly long texts which were not present in the Dev Set – corresponding to Web documents added to the 2010 collection. Considering that the results are similar to the performance of systems in the KBP Slot Filling Task (Median F1: 0,1413), it appears to be worth investigating this validation task further, allowing RTE systems to better tune for the exercise and train to deal with larger amount of data.

5 System Approaches

Seventeen out of eighteen participants in RTE-6 submitted a report on their systems. A first analysis confirms the tendency to address the textual entailment task exploiting Machine Learning techniques (ML). In fact, a large number of participating systems – more than one third - were based on ML, mainly using lexical, syntactic, and semantic features.

A number of other well consolidated techniques were used in RTE-6, such as (i) transformation-based approaches on dependency representations, (ii) methods exploiting similarity measures and/or matching algorithms (based on different levels of analysis - lexical, syntactic, and semantic), and (iii) distance-based approaches.

It is worth noticing that unlike in the previous challenges no logical approaches were adopted in RTE-6. Finally, one knowledge-based approach exploiting semantics alone was tested in RTE-6, consisting of detecting entailment exclusively through an extensive treatment of named entities.

The new task of performing TE in a corpus also led to some novelties with respect to the tools used by the systems. For example, the necessity of dealing with discourse-related issues increased the use of coreference resolution tools, which were employed in 9 systems out of 17 in RTE-6. Moreover, IR tools were used in this challenge.

5.1 Main and Novelty Task Approaches

In this section we provide a brief overview of the approaches used in this year's challenge.

As said, a significant portion of RTE-7 systems were based on Machine Learning. The *budapestacad* system practices a simple ML approach, based on the extractions of semantic relations to compare propositional content. It uses a tool which create triplets from parse trees of H and T, encoding the relations between verbs and their internal and external arguments, and between noun and modifiers. A triple is created for each verb phrase, then entailment is determined on the basis of the number of triplets occurring both in H and T, using both exact and partial lexical matching. The *DFKI* system represents a robust ML approach, based on a single component which incorporates as much as knowledge sources as possible, and exploits features extracted from the output of a dependency parser to create representations of H and T to measure their similarity. The classification is made using ML, not including the voting mechanism about whether to apply a main or fallback strategy used in previous versions of the systems. The *JU_CSE_TAC* system is a SVM machine that uses lexical and syntactic similarity, and rules based on chunking and named entity recognition modules. The strategy used consisted in exploiting both lexical and syntactic modules, which produced up to twenty-five features to be fed to the SVM for entailment decision making. The *Sagan* system consists of a super-

vised ML approach, working almost only on a number of semantic features based on Word Net. For the Main task, the system used feature vectors produced on different versions of the RTE-4 sets, and semantic measures based on Word Net to obtain maximum similarities between concepts. Also the *SINAI* system uses a supervised ML process, and is made up of two modules to extracts features, based on distance between PPVs and matching between named entities in T and H, which are used to train the SVM models. The *SJTU_CIT* uses a machine learning approach centered on structure prediction, where different linguistic resources define features used by a structural SVM to predict semantic alignment and determine entailment. Finally, the *UB.dmirg* system learns entailment relations using lexical resources to extract lexical, semantic parse-free event-based features, employed to determine term matching. A ML module then classifies existing relations between T and H.

A number of approaches other than ML were also used in the RTE-6 challenge. The *BIU* system is a transformation-based TE system, which uses various types of entailment knowledge. The system's architecture is open and enable to experiment with the system and extend it with new inference components. Instead of the supervised learning mechanism used in the past, in RTE-6 BIU exploited a syntax-based matching algorithms to assess the degree of syntactic match between H and T. The system uses also a number of lexical and lexical-syntactic rules to cope with semantic variability, and an IR-based retrieving component to find candidate entailing sentences. The *PKTUM* system is based on surface techniques of lexical and syntactic analysis using semantic knowledge resources. The method consists of practicing transformations over the dependency tree of H using different types of entailment rules, on the assumption that if T entails H, than H can be transformed into T. After transformations are applied to the H dependency tree, this is matched with the representation of the T dependency tree for entailment decision, where high matching indicates semantic relation. Thresholds based on the Development set are also applied. The *UAIC* system is based on word mapping performed by transforming the H through semantic knowledge resources. Its main module maps all the nodes from the H dependency tree to at least one node in T dependency tree, either directly or indirectly. It calculates fitness values on the basis of the transformations needed, indicating the similarity between T and H.

The *FBK_irst* system EDITS performs a distance-based approach, which measures the distance between T and T as the overall cost of the edit operations (insertion, deletion, substitution). The system consists of three modules, namely an edit distance algorithm, a cost scheme, and a set of entailment/contradiction rules, which provide specific knowledge, either lexical, syntactic or semantic. Different configurations where tested in different runs, all based on a learned model by using word-overlap algorithm, without stopword filtering and lexical entailment rules.

The *Boeing* systems, called BLUE-Lite, practices a knowledge-based lexical approach, which does not employ any kind of syntactic information in entailment decision taking, but only a lexical representation of sentences, consisting in bags of words. The comparison between the bags of words generated for T and H takes advantages also from linguistic and world knowledge resources. Moreover, to solve coreference in T, the preceding sentence in the corpus is also taken in consideration as a context. Finally, in deciding entailment, varied thresholds are used for different topic. The *deb_iitb* is a lexical –based system, at whose core there is a matching modules. It utilizes different knowledge resources, and also a coreference module, in order to establish lexical matching between T and H, and a different thresholds for entailment decision. For the Search task a specific method was employed, where in solving coreference also two or three previous sentences were fed together with the candidate T sentence. The *IKOMA* system applies a similarity based method, enhanced by a preliminary processing called "local-novelty" detection. Basically, first a module determines whether H is "local-novel", i.e. determines if the content of H was published for the first time in a determined entailing T, implying that all the T's pre-dating that T do not entail H. The novelty detection module output contributes to the determination of similarity threshold, allowing to set it not to high, in the attempt to raise Precision while minimizing the decline of Recall. The

saicnlp system proposes a rule-based analysis approach, involving deep parsing and propositional pattern matching for entailment determination. After extracting atomic propositions from T, a proposition matching algorithm searches which among those propositions matches H – i.e. which propositions contain a synonymous predicate and a similar set of arguments with respect to H. The *Sangyan* system exploits Dependency Tree Matching to recognize syntactic similarity. At its core it has a syntactic module, which verifies whether similar words play similar syntactic roles, and a semantic module which tries to establish semantic relatedness, if the syntactic model cannot determine entailment. Finally, the *UIUC* systems aims to determine if semantics alone can accurately detect entailment, and uses a knowledge-based approach to identify people, places and organizations. First it performs named entity recognition; then a matching of non-stopwords in H and T; and finally assigns scores to named entities on the basis of their occurrences in both H and T. Entailment is determined considering the overall named entities scores and the number of the remaining non-stopwords in T and H which do not match - a high number indicating a higher probability of no entailment.

5.2 KBP Validation Task Approaches

Three teams participated in the KBP Validation Task, all using the same approach adopted for the Main Task. While the *BIU* system was applied to the KBP Validation data "as-is" (i.e. tuned as for the Main Task), the other two teams exploited ad-hoc configurations. The *JU_CSE_TAC* system utilized two methods, one exploiting lexical similarity, the other an IR technique using Lucene. For the Tailored subtask, a series of validation rules for each attribute were produced using the Development set. *FBK_irst* obtained the learned model by using the edit distance algorithm, without stop-words filtering, using Wikipedia lexical entailment rules. In the two submitted runs, different filtering strategies to reduce the search space were also employed, discarding T-H pairs with low relatedness.

6 RTE-6 Ablation Tests and RTE Knowledge Resources initiative

The exploratory effort started in RTE-5 to perform resource evaluation through ablation tests was not only reiterated in the RTE-6, but also extended to tools. Participants in the Main task were required to carry out ablation tests on all the knowledge resources and tools used in their systems, in order to collect data to better understand the impact of both knowledge resources and tools used by RTE systems and evaluate their contribution to systems' performance. The kind of ablation tests required in the RTE-6 Main Task consists in removing one module at a time from a system, and re-running the system on the test set with the other modules. By comparing these results to those achieved by the complete system, the practical contribution of the individual component can be assessed.

There was a good response to this task. Out of 18 participants in the Main task only one did not submit any ablation tests, because the architecture of the system did not allow the removal of any components. In total, 78 ablations tests were performed and submitted. Despite these guidelines, 20 submitted ablation tests did not specifically ablate knowledge resources or tools, but a variety of other system components, such as entailment algorithms, empirically estimated thresholds, and other statistical features. In one case, a combination of different components was removed from the system instead of a single one.

It is worth mentioning that this year also some systems that had not used any resources or tools (as they worsened the performance on the Dev Set) decided to take part in the evaluation effort by submitting additional runs where one resource or tool was added to the system. These tests can be considered ablation tests, as only one knowledge resource or tool is evaluated, with the difference that the system on which the resource/tool ablation was performed is not the system participating in the Main Task.

Results for all the submitted ablation tests are in the Notebook Appendix. Table 7 gives summary information about the 58 ablation tests complying with our requirements.

For knowledge resources, 46 ablation tests were carried out on a total of 9 different resources.

	Ablated resource # Ablation tests		Impact on systems					
			Positive			Negative		
			# Tests	F1 Range	F1 Average	# Tests	F1 Range	F1 Average
Knowledge Resources	WordNet	**22**	**14**	0.03 18.28	7.54	**8**	-0.02 -3.21	-1.36
	VerbOcean	**7**	**5**	0.07 2.5	1.28	**2**	-4 -1.15	-2.58
	Wikipedia	**6**	**4**	1.08 4.7	2.25	**2**	-3.58 -23.91	-13.75
	FrameNet	**3**	**-**	-	-	**3**	-0.1 -1.84	-1.25
	DIRT	**4**	**1**	3.97	-	**3**	-0.72 -1.56	-1.09
	CatVar	**1**	**1**	0.63	-	**-**	-	-
	Synonym/ Acronym Dictionaries	**1**	**-**	-	-	**1**	-0.76	-
	Dependency Similarity Dictionary	**1**	**-**	-	-	**1**	-13.56	-
	Proximity Similarity Dictionary	**1**	**-**	-	-	**1**	-7.79	-
Tools	Coreference Resolver	**3**	**1**	0.17	-	**2**	-0.88 -1.54	-1.21
	Named Entities Recognition	**5**	**4**	2.22 20.25	10.98	**1**	-1.23	-
	POS tagger	**1**	**1**	4.99	-	**-**	-	-
	Parser	**1**	**-**	-	-	**1**	-1.76	-
	Name Normalization	**1**	**1**	0.65	-	-	-	-

Table 7. Ablated knowledge resources

For tools, 11 ablation tests were performed to evaluate 5 types of tools. For each ablated component, the number of ablation tests submitted is shown, together with the number of runs showing a negative or positive impact of the resource/tool on the system performance. Note that in this year's ablation tests no ablated component had zero impact.

As already noticed in RTE-5, although the data provided by the ablation tests are interesting and give an idea of the contribution of a single component to the performance of a specific system, determining the actual impact of these knowledge resources and tools in general is not straightforward, as systems use the same resource in different ways, and hence the results of the ablation tests are not fully comparable. In fact, as can be seen in the Table 7, some very common resources such as WordNet, VerbOcean and Wikipedia had a positive impact on some systems and a negative impact on others.

Overall, the RTE community is still in the process of learning how to best utilize knowledge resources within the Textual Entailment exercise. An analysis of the ablation test data (available in the Appendix) gives the opportunity to identify which resources were more useful within each systems, and may help learn from the systems which performed well in the ablation tests how to effectively use such resources.

Continuing the initiative started in RTE-5 to promote the study of the impact of knowledge resources on Textual Entailment, the results of all the ablation tests will be made available on the RTE Knowledge Resources web page [10], which already presents last year's results. Beside the ablation test data, the RTE Knowledge Resources page lists the "standard" knowledge resources that have been selected and exploited in the design of RTE systems during the challenges held so far. Currently 35 publicly available and 14 non-publicly available resources are listed. Participants are encouraged to help keep the page up-to-date, sharing their own knowledge resources and tools, not only to contribute to the research on the impact of knowledge resources on RTE, but also to have the opportunity to further test and leverage such resources and tools.

7 Conclusions and Future Work

RTE-6 was characterized by a major innovation, namely the transition from the traditional Main Task, proposed in the first five RTE challenges, to a new Main Task, similar to last year's Pilot Search Task. The main reason behind this choice was the feeling that most RTE systems were ready to make a definitive move towards performing Textual Entailment against a corpus. This change implied an important advantage of the new setting over the Main tasks in the previous challenges, as far as distribution of entailment is concerned. In fact, in the methodology previously used to create the Main data set, T-H examples were picked by hand, and their distribution was artificially balanced (50% YES and 50% NO). In the new setting, the vast majority of Ts in the corpus – selected as candidate entailing sentences through a preliminary Information Retrieval filtering phase – were annotated, thus obtaining a good approximation of the true distribution of entailing sentences in the text, which implies a small – and realistic – proportion of entailments.

The decision to introduce the new main task was rewarded by a good response from the system developers. Even though the results are not completely comparable to those achieved in the RTE-5 Search Pilot, a notable improvement was seen, especially considering the system performances over the Information Retrieval baseline, with the F-measure median being above it – while last year the baseline was above the best result – and with the best result being more than 13 points higher.

Also the experimental Novelty Detection Subtask was quite successful, and the results showed that RTE systems have high performance in detecting novelty, and could be useful for Summarization systems.

In contrast, the KBP Validation Pilot appeared to be more challenging, and only three participants took part in the experiment. A first analysis indicates that one discouraging factor may have been the difference between Development and Test set, as the latter contained a higher number of significantly longer texts taken from the Web. The lack of adequate training and the difficulty that current RTE systems have to process large amount of data made the exercise particularly hard.

Overall, the RTE-6 tasks were interesting and useful for advancing the state of the art in Textual Entailment, and suggest that entailment systems could play an important and effective role within semantic applications. This represents an important step towards the ultimate goals of the RTE challenges, namely assessing Textual Entailment technology, promoting further development in the state of the art, and demonstrating its applicability and usefulness in real-life application scenarios.

Acknowledgments

We would like to acknowledge other people involved in the RTE-6 challenge: Medea Lo Leggio, Alessandro Marchetti, Giovanni Moretti from CELCT, and Karolina Owczarzak from NIST.

This work is supported in part by the Pascal-2 Network of Excellence, ICT-216886-NOE.

References

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini and Idan Szpektor. 2006. The Second PASCAL Recognizing Textual Entailment Challenge. In *Proceedings*

[10] http://www.aclweb.org/aclwiki/index.php?title=RTE_Knowledge_Resources.

of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment, Venice, Italy.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, Medea Lo Leggio and Bernardo Magnini. 2009a. Considering Discourse References in Textual Entailment Annotation. In *Proceedings of the 5th International Conference on Generative Approaches to the Lexicon (GL 2009)*, Pisa, Italy.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, Bernardo Magnini. 2009b. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *TAC 2009 Workshop Notebook*, Gaithersburg, Maryland, USA.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In Quiñonero-Candela, J.; Dagan, I.; Magnini, B.; d'Alché-Buc, F. (Eds.), *Machine Learning Challenges*. Lecture Notes in Computer Science, Vol. 3944, Springer.

Hoa Trang Dang and Karolina Owczarzak. 2009. Overview of the TAC 2009 Summarization Track. In *TAC 2009 Workshop Notebook*, Gaithersburg, Maryland, USA.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, Bill Dolan. 2007. The Third PASCAL Recognizing Textual Entailment Challenge, In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic.

Danilo Giampiccolo, Hoa Trang Dang, Bernardo Magnini, Ido Dagan, Elena Cabrio. 2008. *The Fourth PASCAL Recognizing Textual Entailment Challenge. In TAC 2008 Proceedings*. http://www.nist.gov/tac/publications/2008/papers.html

Joseph L. Fleiss. 1971. *Measuring nominal scale agreement among many raters.* In Psychological Bulletin, 76(5).

Paul McNamee, Hoa T. Dang, (2009). Overview of the TAC 2009 Knowledge Base Population Track. In *Proceedings of the TAC Workshop*, Gaithersburg, MD, USA.

Anselmo Peñas, Alvaro Rodrigo, V. Sama, Felicia Verdejo, (2007). Testing the Reasoning for Question Answering Validation. In *Journal of Logic and Computation*
http://logcom.oxfordjournals.org/cgi/reprint/exm072

Sidney Siegel and N. John Castellan. 1988. *Nonparametric Statistics for the Behavioral Sciences*, McGraw-Hill, New York.