Learning to Link with Wikipedia

David Milne Ian H. Witten

Department of Computer Science, University of Waikato
Private Bag 3105, Hamilton, New Zealand
+64 7 838 4021

{dnk2, ihw}@cs.waikato.ac.nz

ABSTRACT

This paper describes how to automatically cross-reference documents with Wikipedia: the largest knowledge base ever known. It explains how machine learning can be used to identify significant terms within unstructured text, and enrich it with links to the appropriate Wikipedia articles. The resulting link detector and disambiguator performs very well, with recall and precision of almost 75%. This performance is constant whether the system is evaluated on Wikipedia articles or "real world" documents.

This work has implications far beyond enriching documents with explanatory links. It can provide structured knowledge about any unstructured fragment of text. Any task that is currently addressed with bags of words—indexing, clustering, retrieval, and summarization to name a few—could use the techniques described here to draw on a vast network of concepts and semantics.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *text analysis.*

I.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *linguistic processing.*

General Terms

Algorithms, Experimentation.

Keywords

Wikipedia, Data Mining, Semantic Annotation, Word Sense Disambiguation.

1. INTRODUCTION

Wikipedia has seen a meteoric rise in scale and popularity over the last few years. It is now the largest, most visited encyclopedia in existence. It is also densely structured; its articles are peppered with hundreds of millions of links. These connections explain the topics being discussed, and provide an environment where serendipitous encounters with information are commonplace. Anyone who has browsed Wikipedia has likely experienced the feeling of being happily lost, browsing from one interesting topic to the next and encountering information that they would never have searched for. Wikipedia is a classic "small world," so richly hyperlinked that it takes, on average, just 4.5 clicks to get from one article to any other (Dolan, 2008).

The work described in this paper aims to bring the same explanatory links—and the accessibility and serendipity they provide—to all documents. It explains how the topics mentioned in unstructured text can be automatically recognized and linked to the appropriate Wikipedia articles to explain them. Figure 1 illustrates this with a somewhat dated news story about Iranian prisoners of war left in Iraq after the first Gulf War, which has been automatically augmented using our techniques with links to pertinent topics such as the *International Committee of the Red Cross* and *Baghdad*. This process is known as *wikification*, and our approach differs from previous attempts in that we use Wikipedia not only as a source of information to point to, but also as training data for how best to create links. This gives large improvements in both recall and precision.

Before describing the details of this new machine-learning approach to wikification, we first describe the related work to which it can be compared. This is followed by descriptions of the two separate stages involved: link disambiguation and link detection. Both of these steps are evaluated separately against manually defined ground-truth obtained from Wikipedia. This is followed by a third evaluation, in which news stories are wikified and then judged by human participants. The paper concludes with a discussion of implications, which go much beyond enriching documents with explanatory links. The techniques described here can provide structured knowledge about any unstructured fragment of text, and are therefore applicable to a wide variety of tasks.

2. RELATED WORK

Automatically augmenting text with links to web pages has been controversial in the past. When developing Windows XP, Microsoft released plans for the Smart-Tag service which was to automatically add links to web-pages within Windows Explorer. This was aborted when many expressed concern that pages were being "surreptitiously" modified for commercial purposes (Mossberg, 2001). Google's AutoLink feature has received similar criticism and has not been widely accepted. Consequently automatic linking is most successful when restricted to safe domains such as cinema (Drenner *et al.* 2006).

Using Wikipedia as a destination for links sidesteps most of the concerns about automatic link generation, since the resource strives to be impartial and does not generate profits. To our knowledge, the only existing attempt to use Wikipedia in this way is the Wikify system developed by Mihalcea and Csomai (2007). This system works in two separates stages. The first, *detection,* involves identifying the terms and phrases from which links should be made.

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.

Mihalcea and Csomai's most accurate approach to this is based on link probabilities obtained from Wikipedia's articles. Formally, the link probability of a phrase is defined as the number of Wikipedia articles that use it as an anchor, divided by the number of articles that mention it at all. Thus the detection approach is to gather all n-grams for a document and retain those whose link probability exceeds a certain threshold. When tested on Wikipedia articles, the resulting anchor vocabularies matched the original markup with a precision of 53% and a recall of 56%.

The next phase, *disambiguation*, ensures that the detected phrases link to the appropriate article. For most anchors, there are several destinations to choose from. The term *plane*, for example, usually links to an article about fixed wing aircraft. Sometimes, however, it points to a page describing a theoretical surface of infinite area and zero depth, or a tool for flattening wooden surfaces. To choose the most appropriate destination, Wikify's best approach extracts features from the phrase and its surrounding words (the terms themselves and their parts of speech), and compares this to training examples obtained from the entire Wikipedia. When run over anchors obtained from Wikipedia articles, this is able to match the manually defined destinations with a precision of 93% and a recall of 83%. However, it requires enormous preprocessing effort, because the entire Wikipedia must be parsed.

The problem of *topic indexing* is closely related to wikification. Here the aim is to identify the most significant topics; those which the document was written about (Maron, 1977). These index topics can be used to summarize the document and organize it under category-like headings. Wikipedia is a natural choice as a vocabulary for obtaining index topics, since it is broad enough to be applicable to most domains. To use Wikipedia in this way, one must go through much the same process as wikification: one must detect the significant terms being mentioned, and disambiguate these to the appropriate topics. The only difference is an additional stage where the most important topics are identified.

Medelyan *et al.* (2008) make these similarities very clear in their approach to topic indexing with Wikipedia, and even reuse Wikify's approach for detecting significant terms. They differ in how they disambiguate terms, however. They gain similar results much more cheaply by balancing (a) the commonness (or prior probability) of each sense and (b) how the sense relates to its surrounding context. This approach explained in Section 3.1, where we improve upon it by weighting context terms and using machine learning to balanced commonness and relatedness.

3. LEARNING TO DISAMBIGUATE LINKS

This section describes and evaluates a new approach to disambiguating terms that occur in plain text, so they can be linked to the appropriate Wikipedia article. It seems odd to cover this problem first when the techniques described previously tackle the task of *detection*—recognizing terms that should be linked—before deciding where they should link to. This reflects one of the key differences of our approach: it uses disambiguation to inform detection, and thus this stage must be described first.

3.1 A learning approach to disambiguation

We have developed a machine-learning approach to disambiguation that uses the links found within Wikipedia articles for training. For every link, a Wikipedian has manually—and probably with some effort—selected the correct destination to represent the intended sense of the anchor. This provides millions of manually-defined ground truth examples to learn from.

All the experiments described in this paper are based on a version of Wikipedia that was released on November 20, 2007. It contains just under two million articles. Because we wanted a reasonable number of links to use for both training and evaluation, we selected articles containing at least 50 links. We also avoided lists

Figure 1: A news story that has been automatically augmented with links to relevant Wikipedia articles

and disambiguation pages, because these are not representative unstructured text. A total of 700 articles were randomly selected and set aside for developing the disambiguation algorithm: 500 for training; 100 for configuration, and a further 100 for final evaluation.

The 500 training articles contain more than 50,000 links. Each link represents several training instances. The connection between an anchor term and its chosen destination gives a positive example, while the remaining possible destinations provide negative ones. Figure 2 demonstrates this with the anchor *tree*: there are 26 possible senses (18 more than are shown in table on the right). Only one sense is a positive example, and the remaining 25 are negative. In all, the 500 training articles provide about 1.8 million examples.

Commonness and Relatedness

Just like Medelyan et al's (2008) algorithm, our basic approach is to balance the commonness (i.e. prior probability) of a sense with its relatedness to the surrounding context. The commonness of a sense is defined by the number of times it is used as a destination in Wikipedia: Figure 2 shows that 93% of *tree* anchors link to the woody plant, 3% to the type of graph, and 3% to the computer science concept. The algorithm is predisposed to select the first of these senses rather than the more obscure ones, which go all the way down to *The Trees*, a song by the British rock band *Pulp*.

As figure 2 demonstrates, this is not always the best decision. Here *tree* clearly refers to one of the less common senses—the hierarchical data structure—because it is surrounded by computer science concepts. Our algorithm identifies these cases by comparing each possible sense with its surrounding context. This is a cyclic problem because these terms may also be ambiguous. Fortunately in a sufficiently long piece of text one generally finds terms that do not require any disambiguation at all, because they are only ever used to link to one Wikipedia article. There are four unambiguous links in the text of Figure 2, including *algorithm*, *uninformed search* and *LIFO stack*. We use every unambiguous link in the document as context to disambiguate ambiguous ones.

Each candidate sense and context term is represented by a single Wikipedia article. Thus the problem is reduced to selecting the sense article that has most in common with all of the context articles. Comparison of articles is facilitated by the Wikipedia Link-based Measure we developed in previous work (Milne and Witten, 2008), which measures the semantic similarity of two Wikipedia pages by comparing their incoming and outgoing links. For the sake of efficiency the disambiguation algorithm (and the link detection system that follows) only considers the links made to each article. The algorithm must make a vast amount of comparisons, and this small sacrifice allows all of the information required to do so to be stored in memory. Formally, the relatedness measure is:

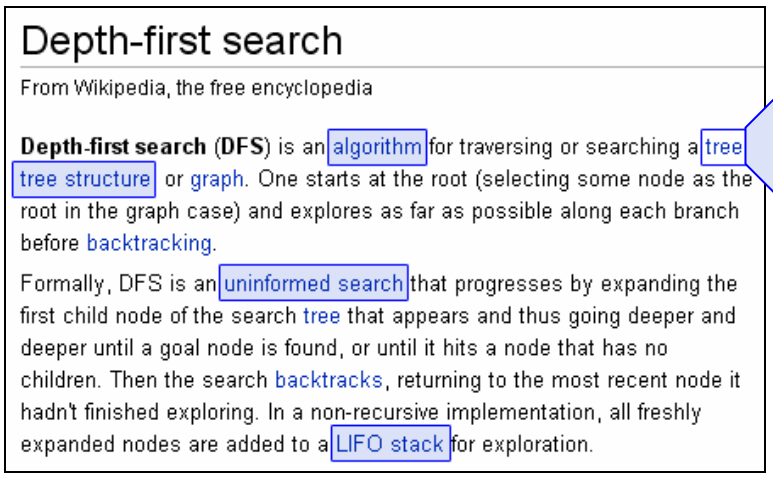
$$relatedness(a,b) = \frac{\log(\max(|A|,|B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|,|B|))}$$

where a and b are the two articles of interest, A and B are the sets of all articles that link to a and b respectively, and W is set of all articles in Wikipedia. The relatedness of a candidate sense is the weighted average of its relatedness to each context article, where the weight of each comparison is defined in the next section.

Some context terms are better than others

One of the main differences between our approach and Medelyan et al's is that we do not consider all context terms to be equally useful. The word *the*, for example, is unambiguous in that it is only ever used to link to the grammatical concept of an article, but it has zero value for disambiguating other concepts. Mihalcea and Csomai's *link probability* feature helps to identify such cases; there are millions of articles that mention *the* but do not use it as a link. Weighting context terms on this feature emphasizes those that are most likely a priori—ones that are almost always used as a link within the articles where they are found, and always link to the same destination.

Secondly, many of the context terms will be outliers that do not relate to the central thread of the document. We can determine how closely a term relates to this central thread by calculating its average semantic relatedness to all other context terms, using the measure described previously. These two variables—link probability and relatedness—are averaged to provide a weight for each context term. This is then used when calculating the weighted average of a candidate sense to the context articles.



sense	commonness	relatedness
Tree	92.82%	15.97%
Tree (graph theory)	2.94%	59.91%
Tree (data structure)	**2.57%**	**63.26%**
Tree (set theory)	0.15%	34.04%
Phylogenetic tree	0.07%	20.33%
Christmas tree	0.07%	0.0%
Binary tree	0.04%	62.43%
Family tree	0.04%	16.31%
…		

Figure 2: Disambiguating tree *using surrounding unambiguous links as context.*

	recall	precision	f-measure
Naïve Bayes	96.6	95.0	95.8
C4.5	96.8	**96.5**	96.6
Support Vector Machines	96.5	96.0	96.3
Feature selected C4.5	96.8	**96.5**	96.6
Bagged C4.5	**97.3**	**96.5**	**96.9**

Table 1: Performance of classifiers for disambiguation over development data

	recall	precision	f-measure
Random sense	56.4	50.2	53.1
Most common sense	92.2	89.3	90.7
Medelyan *et al.* (2008)	92.3	93.3	92.9
Most valid sense	95.7	**98.4**	**97.1**
All valid senses	**96.6**	97.0	96.8

Table 2: Performance of disambiguation algorithms over final test data

Combining the features

We have already discussed the two main features used by the classifier: the commonness of each sense, and its relatedness to the surrounding context. Only the latter of these is different from those used by Medelyan *et al*. A more fundamental difference is the way in which we use machine learning to combine these features, so that the balance can be adjusted from document to document. The previous work instead used a fixed prior heuristic, determined in advance.

To balance commonness and relatedness, we take into account how good the context is. If it is plentiful and homogenous, then relatedness becomes very telling. In Figure 2, for example, the most common sense of *tree* is entirely irrelevant because the document is clearly about computer science. However, if *tree* is found in a general document with ambiguous or confused context, then the most common sense should be chosen. By definition, this will be correct in most cases. Thus the final feature—context quality—is given by the sum of the weights that were previously assigned to each context term. This takes into account the number of terms involved, the extent they relate to each other, and how often they are used as Wikipedia links.

The three features are used to train a classifier that can distinguish valid senses from irrelevant ones. It does not actually choose the best sense for each term. Instead it considers each sense independently, and produces a probability that it is valid. If strict disambiguation is required, then we simply choose the sense that has the highest probability. If more than one sense may be useful, then we gather all senses that have a higher probability of being valid than not. We evaluate these options in Section 3.3.

3.2 Configuration and attribute selection

Configuring the disambiguation classifier involves setting one parameter and identifying the most suitable classification algorithm. This parameter specifies the minimum probability of senses that are considered by the algorithm. As illustrated earlier with the *tree* example, terms often have extremely unlikely senses which can be safely ignored. The distribution follows the power law: the vast majority of links are made to just a few destinations and there is a long tail of extremely unlikely senses. *Jackson*, for example, has 230 senses, of which only 31 have more than 1% chance of occurring. If all these are considered they must each be compared to all the context terms. Much speed is gained by imposing a threshold below which all senses are discarded. This has the added advantage of increasing precision, since the discarded senses are unlikely to be relevant, but it decreases recall. Figure 3 plots this tradeoff, and identifies 2% as a sensible probability threshold that balances the two metrics.

We experimented with several classification algorithms, and the results are shown in Table 1. As one would expect, Naïve Bayes has the worst performance. There are dependencies between the features that lead this scheme astray. Interestingly Quinlan's (1993) C4.5 algorithm outperforms the more sophisticated Support Vector Machine, and so it is used in the remainder of the paper. Feature selection makes no difference, and bagging improves the classifier by only 0.3%.

3.3 Evaluation

To evaluate the disambiguation classifier, 11,000 anchors were gathered from 100 randomly selected articles and disambiguated automatically. Table 2 compares the result with three baselines. The first chooses a *random sense* from the anchor's list of destinations. Another always chooses the *most common sense.* The final baseline is the heuristic approach developed by Medelyan *et al.* (2008).

Having our classifier choose what it considers to be the *most valid sense* for each term outperforms all other approaches. The key differences between this and Medelyan *et al*.'s system are the use of machine learning and the weighting of context. These provide a 76% reduction in error rate. The classifier never gets worse than 88% precision on any of the documents, and for 45% of documents it attains perfect precision. Recall is never worse than 75%, and perfect for 14% of documents. Recall can be increased by allowing the classifier to select *all valid senses*. Unfortunately this causes precision to degrade and makes for slightly lower overall performance. Consequently strict disambiguation is used throughout the remainder of this paper.

Mihalcea and Csomai's best disambiguation technique had an f-measure of 88%. Direct comparison may not be fair, however, since their disambiguation approach was evaluated on an older version of Wikipedia. One could argue that the task gets more difficult over time as more senses (Wikipedia articles) are added, in which case it is encouraging that our approach (which was run

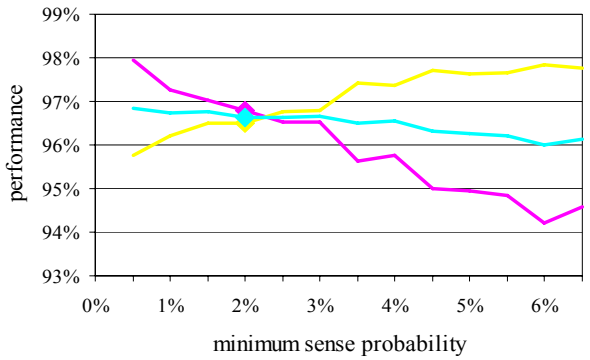


Figure 3: Disambiguation performance vs. minimum sense probability.

on newer data) gains better results. On the other hand disambiguation may well be getting easier over time. The baseline of simply choosing the most common senses has improved since Mihalcea and Csomai's experiments, which shows that common senses are becoming more and more dominant. Consequently any algorithm that is trained and tested on the newer documents will inherently have a higher accuracy. In any case, our approach is competitive and has a distinct advantage of not requiring parsing of the text. This significantly reduces the resources required and, in principle, provides language independence. Additionally the system requires much less training (500 articles vs. the entire Wikipedia). On a modest desktop machine (with a 3Ghz Dual Core processor and 4Gb of RAM) the new disambiguator was trained in 13 minutes and tested in four, after spending another three minutes loading the required summaries of Wikipedia's link structure and anchor statistics into memory.

This evaluation can also be considered as a large-scale test of our Wikipedia link-based measure. Just the testing phase of the experiment involved more than two million comparisons in order to weight context articles and compare them to candidate senses. When these operations were separated out from the rest of the disambiguation process they where performed in three minutes (a rate of about 11,000 every second) on the desktop machine.

4. LEARNING TO DETECT LINKS

This section describes a new approach to link detection. The central difference between this and Mihalcea and Csomai's system is that Wikipedia articles are used to learn what terms should and should not be linked, and the context surrounding the terms is taken into account when doing so. Wikify's detection approach, in contrast, relies exclusively on link probability. If a term is used as a link for a sufficient proportion of the Wikipedia articles in which it is found, they consider it to be a link whenever it is encountered in other documents—regardless of context. This approach will always make mistakes, no matter what threshold is chosen. No matter how small a terms link probability is, if it exceeds zero then, by definition, there is some context in which has been used as a link. Conversely, no matter how large the probability is, if it is less than 1 there is some context where it should not be used a link. Thus this approach will always discard relevant links and retain irrelevant ones, regardless of chosen threshold. We are able to gain much better results by only using link probability as one feature among many.

4.1 A machine-learning link detector

The link detection process starts by gathering all n-grams in the document, and retaining those whose probability exceeds a very low threshold. This threshold—the value of which is established in the next section—is only intended to discard nonsense phrases and stop words. All the remaining phrases are disambiguated using the classifier described in the previous section. As shown in Figure 4, this results in a set of associations between terms in the document and the Wikipedia articles that describe them, which is obtained without any form of part-of-speech analysis. Sometimes, as is the case with *Democrats* and *Democratic Party,* several terms link to the same concept if that concept is mentioned more than once. Sometimes, if the disambiguation classifier found more than one likely sense, terms may point to multiple concepts. *Democrats*, for example, could refer to the party or to any proponent of democracy.

These automatically identified Wikipedia articles provide training instances for a classifier. Positive examples are the articles that were manually linked to, while negative ones are those that were not. Features of these articles—and the places where they were mentioned—are used to inform the classifier about which topics should and should not be linked. The features are as follows.

Link Probability. Mihalcea and Csomai's link probability is a proven feature. On its own it is able to recognize the majority of links. Because each of our training instances involves several candidate link locations (e.g. *Hillary Clinton* and *Clinton* in Figure 4), there are multiple link probabilities. These are combined into two separate features: the average and the maximum. The former is expected to be more consistent, but the latter may be more indicative of links. For example, *Democratic*

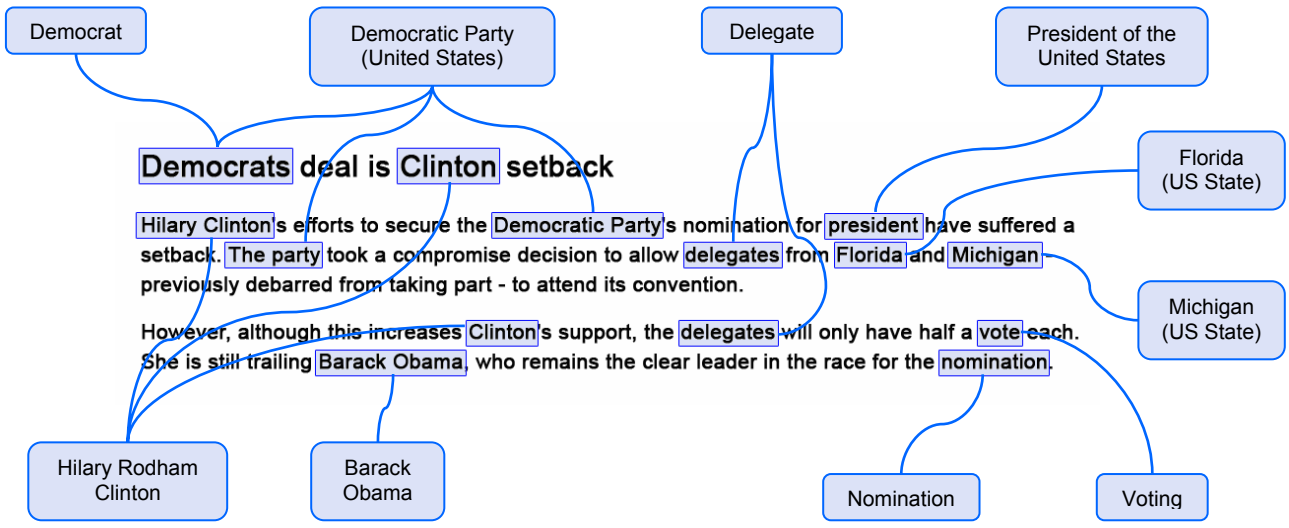


Figure 4: Associating document phrases with appropriate Wikipedia articles

Party has a much higher link probability than *the party*. As a matter of style, this document only refers to it once by its proper name. The fact that it was important enough to be referred to in full is a strong indication of link-worthiness, but this information is lost when the probabilities are averaged.

Relatedness. Intuitively, one would expect that topics which relate to the central thread of the document are more likely to be linked. *Clinton, Obama,* and the *Democratic Party* are more likely to be of interest to the reader than *Florida* or *Michigan*. Recall that we have already gone to some lengths to obtain relatedness score between each topic and its surrounding context, in order to disambiguate them. This provides a relatedness feature with no further computation. However, since the semantic relatedness comparisons are all but free, we augment this with a second feature: the average relatedness between each topic and all of the other candidates.

Disambiguation Confidence. The disambiguation classifier described earlier does not just produce a yes/no judgment as to whether a topic is a valid sense of a term; it also gives a probability or confidence in this answer. We use this as a feature to give those topics that we are most sure of a greater chance of being linked. As with link probability, there may be multiple confidence values for each instance because several different terms may be disambiguated to the same topic. These are again combined as average and maximum values, for the same reasons.

Generality. It is more useful for the reader to provide links for specific topics that they may not know about, rather than general ones that require little explanation. We define the *generality* of a topic as the minimum depth at which it is located in Wikipedia's category tree. This is calculated beforehand by performing a breadth-first search starting from the *Fundamental* category that forms the root of Wikipedia's organizational hierarchy.

Location and Spread. The remaining features are based on the locations where topics are mentioned; i.e. the n-grams from which they were mined. *Frequency* is an obvious choice, since the more times a topic is mentioned, the more important and link-worthy it is. Another is *first occurrence* because, as observed by David *et al*. (1995), topics mentioned in the introduction of a document tend to be more important. Significant topics are also likely to occur in conclusions, so *last occurrence* is also used. Finally the distance between first and last occurrences, or *spread*, is used to indicate how consistently the document discusses the topic. These last three location-based features are all normalized by the length of the document.

4.2 Training and configuration

As with the disambiguation classifier, we have set aside three different sets of Wikipedia articles for training, configuration and evaluation. The same 500 articles used to train the disambiguation classifier are used for training here. This is done to reduce the number of disambiguation errors, because these directly affect the quality of training. As described earlier, terms must be disambiguated into appropriate articles before they can be used as training instances. If a valid link were disambiguated incorrectly then many of its features would indicate a valid link, but the instance would be a negative example. Reusing the training data reduces the chance these confusing examples occurring.

	recall	precision	f-measure
Naïve Bayes	70.2	70.3	70.2
C4.5	**77.6**	72.2	74.8
Support Vector Machines	72.5	**75.0**	73.7
Bagged C4.5	77.3	72.9	**75.0**

Table 3: Performance of classifiers for link detection

Likewise, configuration is done on the same 100 articles used to configure the disambiguation classifier, simply because there is no reason not to reuse them. The only variable to configure is the initial link probability threshold used to discard nonsense phrases and stop words. This variable sets up a tradeoff with speed and precision on one side and recall on the other, since a higher threshold means only the most likely instances are inspected, but risks discarding valid links. Figure 5 plots this tradeoff, and identifies 6.5% link probability as the point where precision and recall are balanced.

Despite our choice of training data, we found that the disambiguation classifier described in Section 3 performed quite poorly when used as part of the wikification classifier. It became very accepting, considering not just one or two senses to be valid for each term, but five or six. This is because the disambiguator was trained on links, but is being used here on raw text. In training, the context was restricted to manually defined anchors, but here it is mined from all unambiguous terms that have a link probability above the initial threshold. The problem was resolved by modifying the disambiguation training to take these other unambiguous terms into account. The resulting disambiguation classifier was 1% worse (f-measure) when disambiguating links, but behaves more consistently when incorporated into the wikifier.

Table 3 lists the results of the various classifiers we experimented with. Naïve Bayes performs reasonably well since all of the features are fairly independent. Again, C4.5 outperforms support vector machines overall, although the latter attains significantly higher precision. The evaluation described in the next section uses bagged C4.5 in order to gain the best overall results.

4.3 Evaluation

Evaluation of the link detector was performed over an entirely new randomly selected subset containing 100 Wikipedia articles. Ground truth was obtained by gathering the 9,300 topics that these articles were manually linked to. The articles were then

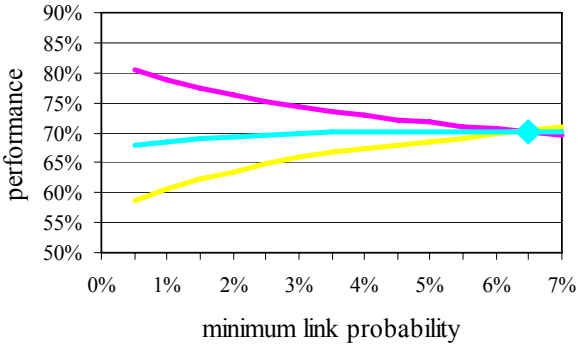


Figure 5: Link detection performance vs. minimum link probability.

	recall	precision	f-measure
Wikify (estimate)	46.5	49.6	48.0
Wikify (upper bound)	53.4	55.9	54.6
New link detector	**73.8**	**74.4**	**74.1**

Table 4: Performance of link detection algorithms

stripped of all markup and handed to the link detector, which produced its own list of link-worthy topics for each article. This evaluation is only concerned with identifying the correct topics that should be linked to, and not the exact locations from which these links should be made. This is consistent with Mihalcea and Csomai's work, which compared vocabularies of anchors, but not their locations.

The result is shown in Table 4, where recall, precision, and f-measure are all approximately 74%. There is a marked drop in performance between disambiguating links and detecting them, but this is to be expected. Deciding where a link should be made to is far less subjective than deciding whether the link should be made at all. The time required is also significantly increased, even though many of the features are carried over from the disambiguator. The link detector was trained in 37 minutes, and tested (while simultaneously performing disambiguation) in 8 minutes.

Wikify's two stages of detection and disambiguation were evaluated individually, but the combined result when both operated together was not reported. In our approach the two stages are inseparable, which makes comparison difficult. Fortunately one can estimate Wikify's overall accuracy by assuming that disambiguation performance is constant across all terms, and combining recall and precision across the two steps as we have done in Table 4. Even if we were to assume perfect disambiguation, the upper bound of this system would be an f-measure of 55%. This shows how our algorithm dramatically improves upon its predecessor. Recall is increased by 59% over the estimate, precision by 50%, and overall f-measure by 54%. As with the previous experiment, a limitation of this comparison is that the two link detection approaches were developed and evaluated on different versions of Wikipedia.

5. WIKIFICATION IN THE WILD

All of the experiments described up until this point have treated Wikipedia as both training ground and proving ground. Even though we have taken steps to ensure that training and testing sets are kept separate, it is still reasonable to wonder whether the process works as well (or at all) on documents that are not obtained from Wikipedia. This section aims to address such concerns by applying our techniques to new documents, and placing the results in front of human evaluators.

5.1 Experimental Data

The test set for this experiment is a subset of 50 documents from the AQUAINT text corpus: a collection of newswire stories from the Xinhua News Service, the New York Times, and the Associated Press. We randomly selected documents from the last of these providers, restricting selection to short documents (250-300 words) to avoid overtaxing the attention spans of the human evaluators.

As training we use another collection of 500 Wikipedia articles. The original intention was to use exactly the same set as the previous experiments, but unfortunately the difference in size between these verbose encyclopedic articles and the short news stories produced a classifier that identified very few link-worthy topics. Consequently we created a new training set by gathering all of the Wikipedia articles of the same length as the newswire stories, and selecting those that contained the highest proportion of links. The resulting classifier identified 449 link-worthy topics within the 50 newswire stories, an average of 9 links per document. Figure 1 shows one of these automatically tagged documents.

5.2 Participants and Tasks

To gather willing participants to inspect the wikified news stories we turned to Mechanical Turk (Barr and Cabrera 2006), a crowd-sourcing service hosted by Amazon. This service provides what Barr and Cabrera describe as *artificial artificial intelligence*; a way for human judgment to be easily incorporated into software applications. From the perspective of the people who develop these applications—who are known as *requestors*—the process is a function call where a question is asked and the answer is returned. What makes this system unique is the thousands-strong crowd of human contributors—or *workers*—who wait at the receiving end of the calls. These people identify the tasks they are interested in, submit their responses, and (pending review) receive payment for their efforts.

For our purposes, Mechanical Turk provided the means to conduct a labor-intensive experiment under significant time constraints, without having to gather participants ourselves. Naturally this raises some concerns about whether the anonymous workers could be trusted to invest the required effort and give well considered responses. Even more alarming, it is possible for Mechanical Turk tasks to be done by automated "bots" created to gather funds for unscrupulous would-be workers (Howe 2006). We implemented several checks to identify and reject low-quality responses and undesirable participants. These are discussed in the following sections, which describe the two different types of tasks that we had the workers perform.

Evaluating detected links

To evaluate the quality of the links that the system produced, we created 449 different tasks; one for each of the links. In each task the evaluator was given the text of the news article exactly as shown in Figure 1, except with only one of the links shown. As is demonstrated for *Baghdad* in the figure, the link was presented with a popup box containing the first paragraph of the relevant Wikipedia article. This allowed both the context of the link and its intended destination to be taken in at a glance. The participant was then given the following options to specify whether the link was valid:

- No - *Baghdad* is not a plausible location for a link.
- No - *Baghdad* is a plausible location, but the link doesn't go to the right Wikipedia article.
- Kind of – *Baghdad* is a plausible link to the correct Wikipedia article, but the article isn't helpful or relevant enough to be worth linking to.
- Yes - *Baghdad* is a plausible link to the correct Wikipedia article, and this article is helpful and relevant.

Only the last option indicates that the link was detected correctly. The other three identify the different reasons why the algorithm made a mistake. The first indicates a term or phrase should not have been considered as a candidate, the second identifies a candidate that was disambiguated incorrectly, and the third indicates a candidate that should have been discarded in the final selection stage. It should be noted that judging the helpfulness and relevance of a link is subjective. In order to do so, participants were asked to put themselves in the shoes of someone who was genuinely interested in the story, and judge whether the linked Wikipedia article would be worthy of further investigation.

To cope with subjectivity and verify individual responses, each task was performed by three different people. To ensure that the task was completed by real participants (rather than bots), each task was paired with a unique completion code that had to be submitted alongside the answer. To ensure that participants gave well considered answers, this code was only made available after the worker had spent at least 30 seconds inspecting the link and its surrounding context. An additional check was to only accept workers who had gained a high reputation from other requestors, by having at least 90% of their responses to previous tasks accepted and rewarded. After rejecting and returning invalid submissions, we eventually gathered responses from 88 different people, who evaluated an average of 15 and a maximum of 156 links each. They spent an average of 1.5 minutes on each link, giving a total of 36 man hours of labor.

Identifying missing links

A second type of task was created to identify the links that our algorithm should have detected, but failed to. In each of the 50 tasks (one for each document) the evaluator was given the news story with all of the detected links clearly identified, exactly as shown in Figure 1. Again each link could be clicked to reveal a popup box that summarized the intended destination. The participants were then asked to list any additional Wikipedia topics that they felt should be linked to, by supplying both the phrase where the link should start from and the URL of the Wikipedia article it should go to. They were asked not to add every single concept that was mentioned, since this is not what wikification aims to do. Instead they were instructed to only choose articles that were relevant for the news article, and which readers would likely to want to investigate further.

We implemented the same checks as before to ensure that the answers were genuine and well-considered. Due to the increased difficulty and subjectivity of these tasks, each was conducted by five different participants, and the minimum time spent on them was increased to five minutes. After rejecting and returning invalid submissions, we eventually gathered responses from 29 different people, who evaluated an average of 8.6 and a maximum of 35 documents each. In total they invested 47 man hours of labor, or an average of 11 minutes on each document.

5.3 Results

As is to be expected for subjective tasks, there was some disagreement between the evaluators. In the case of the first group of tasks this was unfortunately exacerbated by ambiguity. When an evaluator encountered a link that they felt was irrelevant, such as *1980* in Figure 1, they had two equally valid responses available: they could say that the location of the link was implausible, or that the Wikipedia article it pointed to was unhelpful. We resolved this issue by combining the responses in the analysis stage into a single option: that the link was irrelevant and/or unhelpful. Following this combination, we found that 57% of the links received a unanimous decision from all three evaluators. Almost all of the remaining links received a two-vs.-one vote, for which the majority decision was considered correct. 3% of the links received different responses from all of the evaluators. Because there is only one possible response that indicates a valid link, these were judged to be incorrect—for an unknown reason.

correct	76.4
incorrect (wrong destination)	0.9
incorrect (irrelevant and/or unhelpful)	19.8
incorrect (unknown reason)	2.9

Table 5: Accuracy of the automatically detected links.

Table 5 shows the results. Here we see that the precision of the algorithm is 76%, meaning that 34% of the links were incorrect. Almost all of the mistakes were due to incorrect candidate identification or selection, with only four links identified as being incorrectly disambiguated. As mentioned earlier, about 3% of the links were judged differently by all of the evaluators, and thus the reason for their rejection could not be identified.

For the second type of task, the evaluators identified just under 400 distinct Wikipedia articles that they felt were worthy of linking to. This equates to around 8 additional links per document. Because of the subjectivity of the task, the participants did not entirely agree on the articles that were to be added. The majority (53%) of additional links were only identified by one of the participants. 17% were identified by two participants, 13% by three, another 13% by four, and only 4% were unanimously considered to be missing by all five participants. To compile the diverse opinions into coherent judgments, we required that the majority (at least 3) of the participants identify a link before it was considered link-worthy. This produced 117 links that the algorithm should have added to the documents, but didn't.

The results of both sets of tasks were used to correct the original automatically-tagged articles and generate ground truth. The four links that were identified as pointing to the wrong article were manually corrected by the authors. All of the remaining invalid links were simply discarded, and the missing links that were identified by the majority of our participants were added. The result is a new corpus containing only manually-verified links, which we have made available online.[1]

Comparison of the original (automatically tagged) articles with this manually-verified corpus reveals the performance of the topic detector. As mentioned previously, precision is 76%; slightly better than when the system was tested on Wikipedia articles (Table 4). Recall is 73%; just one point worse than in the previous experiment. F-measure is 75%. Overall the figures are remarkably close to those obtained when the system was evaluated against Wikipedia articles, which indicates that algorithm works as well "in the wild" as it does on Wikipedia.

[1] The manually verified and corrected corpus of wikified news articles is available at *www.nzdl.org/wikification*

6. EXAMPLES AND IMPLICATIONS

We have described an algorithm that disambiguates terms to their appropriate Wikipedia articles, and determines those that are most likely to be of interest to the reader. It is easy to imagine applications for this, such as adding explanatory links news stories or educational documents, or detecting missing links in Wikipedia articles and smoothing the process for contributing to them. However, this barely scratches the surface of potential applications.

In essence, we have developed a tool that can accurately cross-reference documents with the largest knowledge base in existence. It can provide structured knowledge about any unstructured document, because it can represent them as graphs of the concepts they discuss. As an illustration of this, Figure 6 shows a sample of topics that were automatically extracted from the content of this paper, where a link between two topics indicates that they have a significant relation between them according to the Wikipedia link-based measure. For clarity sake only a small sample of the relations are shown. *Computer Science*, for example, relates to almost every other topic.

The graph isn't perfect, since it is missing key concepts such as wikification and disambiguation. Nevertheless, it provides a very clear sense of what this paper is about. It resolves ambiguity, so we know exactly what type of *ontology* is mentioned. It does the same for polysemy, so it doesn't matter if the document talked about *knowledge discovery*, *knowledge mining*, *data mining*, or *KDD*—they are all the same thing. By navigating the relationships of meaning between the topics, one can identify the threads of discussion; there is a cluster of topics relating to *ontologies* and *knowledge bases*, another to *natural language processing*, and another to *machine learning*. More formal reasoning can be made available by taking the (trivial) step of connecting to Wikipedia-derived ontologies such as DBPedia (Auer *et al.* 2007), Yago (Suchanek *et al.* 2007), and others (Völkel *et al.* 2006, Ponzetto and Strube 2007). Using these resources one could tell, for example, that *Hamilton* is a city in *New Zealand*, and that it is the home of the *University of Waikato*. All of this adds up to a machine readable representation of the document that is extremely informative.

7. CONCLUSIONS

We are by no means the first to recognize Wikipedia's potential for describing and organizing information. It is fast becoming the resource of choice for such tasks, and has been applied to text categorization (Gabrilovich and Markovich 2007), indexing (Medelyan *et al.* 2008), clustering (Banerjee *et al.* 2007), searching (Milne *et al.* 2007), and a host of other problems. This popularity is entirely understandable: Wikipedia offers scale and multilingualism that dwarfs other knowledge bases, and an ability to evolve quickly and cover even the most turbulent of domains (Lih 2004).

All these applications of Wikipedia face the same hurdle: they must somehow move from unstructured text to a collection of relevant Wikipedia topics. Researchers have discovered many different ways of doing so, but most have not been evaluated independently. Instead these methods are only evaluated extrinsically, by how well they support the overall task.

The present paper's contribution is a proven method of extracting key concepts from plain text that has been evaluated against an extensive body of human performance. This has extraordinarily wide application. Any task that is currently addressed using the bag of words model, or with knowledge obtained from less comprehensive knowledge bases, could benefit from using our technique to draw upon Wikipedia topics instead. We have shown how to reap a bountiful and unexpected new harvest from the countless man-hours that have already been invested by the Web 2.0 community to explain and organize the sum total of human knowledge about our world.

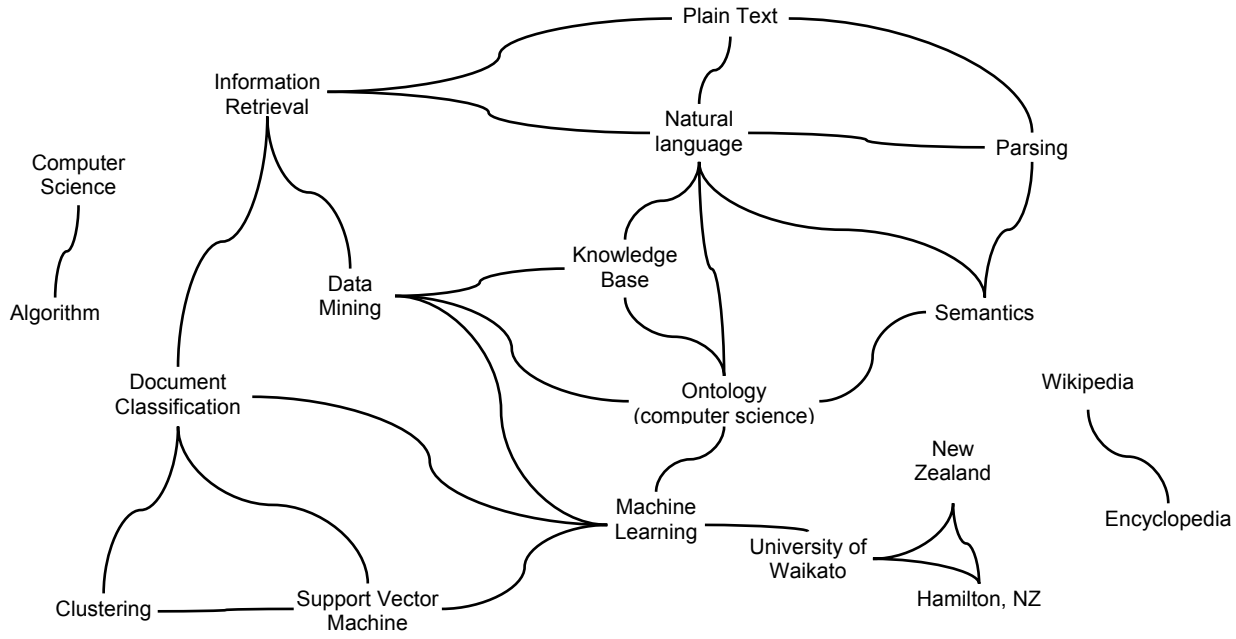


Figure 6: Topics and relations automatically extracted from the content of this paper.

8. ACKNOWLEDGEMENTS

We would like to thank Olena Medelyan and Simone Ponzetto for their ideas and advice, and Rada Mihalcea and Andras Csomai (the authors of the original Wikify system) for sharing data and contributing to the review process. We are also indebted to the WEKA team for producing an excellent resource for machine learning. Finally, we must of course acknowledge the tireless efforts of the Web 2.0 community, without whom resources like Wikipedia and Mechanical Turk would not exist.

This research was conducted with funding from the New Zealand Tertiary Education Commission and the New Zealand Digital Library Group.

9. REFERENCES

[1] Auer, S. and Bizer, C. and Kobilarov, G. and Lehmann, J. and Cyganiak, R. and Ives, Z. (2007) DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International Semantic Web Conference*, Busan, Korea.

[2] Banerjee, S. and Ramanathan, K. and Gupta, A. (2007) Clustering short texts using Wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, Amsterdam, pp. 787-788.

[3] Barr, J. and Cabrera, L.F. (2006) AI gets a brain. In *ACM Queue 4*(4), pp. 24-29.

[4] David, C., L. Giroux, S. Bertrand-Gastaldy, and D. Lanteigne (1995) Indexing as problem solving: A cognitive approach to consistency. In *Proceedings of the ASIS Annual Meeting*, Medford, NJ, pp. 49-55.

[5] Dolan, S. (2008) Six Degrees of Wikipedia. Retrieved June 2008 from *www.netsoc.tcd.ie/~mu/wiki/*

[6] Drenner, S., Harper, M., Frankowski, D., Riedl, J. and Terveen, L. (2006) Insert movie reference here: a system to bridge conversation and item-oriented web sites. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, New York, NY, pp. 951-954

[7] Gabrilovich, E. and Markovitch, S. (2007) Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, Boston, MA.

[8] Howe, J. (2006) The Rise of Crowdsourcing. In *Wired Magazine 14*(6).

[9] Lih, A. (2004) Wikipedia as Participatory Journalism: Reliable Sources? Metrics for evaluating collaborative media as a news resource. In *Proceedings of the 5th International Symposium on Online Journalism*, Austin, Texas.

[10] Maron, M.E. (1977) On indexing, retrieval and the meaning of about. In *Journal of the American Society for Information Science 28*(1), pp. 38-43

[11] Medelyan, O., Witten, I.H. and Milne, D. (2008) Topic Indexing with Wikipedia. In *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008)*, Chicago, IL.

[12] Mihalcea, R. and Csomai, A. (2007) Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the 16th ACM Conference on Information and Knowledge management (CIKM'07),* Lisbon, Portugal, pp. 233-242

[13] Milne, D., Witten, I.H. and Nichols, D.M. (2007). A Knowledge-Based Search Engine Powered by Wikipedia. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM'2007)*, Lisbon, Portugal.

[14] Milne, D., and Witten, I.H. (2008) An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008)*, Chicago, IL.

[15] Mossberg, W. (2001) New Windows XP Feature Can Re-Edit Others' Sites. *The Wall Street Journal*, June 2001

[16] Ponzetto, S.P. and Strube, M. (2007) Deriving a Large Scale Taxonomy from Wikipedia. In *Proceedings of the 22st National Conference on Artificial Intelligence (AAAI'07)*, Vancouver, British Columbia, pp. 1440-1445.

[17] Quinlan, J.R. (1993) *C4. 5: Programs for Machine Learning*. Morgan Kaufmann

[18] Suchanek, F.M. and Kasneci, G. and Weikum, G. (2007) Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web (WWW'07)*, Alberta, Canada, *pp. 697-706.*

[19] Völkel, M. and Krötzsch, M. and Vrandecic, D. and Haller, H. and Studer, R. (2006) Semantic Wikipedia. In *Proceedings of the 15th international conference on World Wide Web (WWW'06)*, Edinburgh, Scotland, pp. 585-594