Natural Language Engineering **20** (4): 469–500. © Cambridge University Press 2013
doi:10.1017/S1351324913000296

Densification: Semantic document analysis using Wikipedia

IUSTIN DORNESCU and CONSTANTIN ORĂSAN
Research Institute in Information and Language Processing, University of Wolverhampton, Wolverhampton, UK
e-mail: {I.Dornescu2, C.Orasan}@wlv.ac.uk

(*Received 10 May 2013; revised 11 September 2013; accepted 13 September 2013; first published online 14 October 2013*)

Abstract

This paper proposes a new method for semantic document analysis: densification, which identifies and ranks Wikipedia pages relevant to a given document. Although there are similarities with established tasks such as wikification and entity linking, the method does not aim for strict disambiguation of named entity mentions. Instead, densification uses existing links to rank additional articles that are relevant to the document, a form of explicit semantic indexing that enables higher-level semantic retrieval procedures that can be beneficial for a wide range of NLP applications. Because a gold standard for densification evaluation does not exist, a study is carried out to investigate the level of agreement achievable by humans, which questions the feasibility of creating an annotated data set. As a result, a semi-supervised approach is employed to develop a two-stage densification system: filtering unlikely candidate links and then ranking the remaining links. In a first evaluation experiment, Wikipedia articles are used to automatically estimate the performance in terms of recall. Results show that the proposed densification approach outperforms several wikification systems. A second experiment measures the impact of integrating the links predicted by the densification system into a semantic question answering (QA) system that relies on Wikipedia links to answer complex questions. Densification enables the QA system to find twice as many additional answers than when using a state-of-the-art wikification system.

1 Introduction

In recent years Wikipedia has become the subject of an increasing number of research projects, in domains such as information extraction, knowledge representation, IR and the Semantic Web. As a repository of encyclopaedic information, Wikipedia has proved to be a very valuable resource for the NLP community because it covers a wide variety of topics and contains a mix of text and semi-structured information. One of the main sources of information from Wikipedia is the abundance of entities that are explicitly annotated in the form of wikilinks, a resource of interest to NLP research. Because articles are meant for human readers, the guidelines for editing articles[1] only allow links to the most relevant articles, to avoid clutter and

[1] http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

redundant information. As a result, the current Wikipedia markup is incomplete, especially when it is to be processed by computers.

This paper studies the problem of identifying and ranking Wikipedia articles relevant to a given text. Such an article, called *target*, is usually referred to by a small span of text called *mention*, *anchor* or *keyword*. The target can be a concept or technical term (e.g. gross domestic product), a general topic (e.g. Health in the United Kingdom), or a named entity, such as people, companies, geographical landmarks, geo-political entities, movies, novels and so on. More specifically, given a textual document d and an encyclopaedic knowledge base comprising a set of entries E, the task is to identify and disambiguate textual references ref_i from d, linking them to related entries e_j in the knowledge base. The task is called *densification* because it increases the density of links in a document by taking advantage of the set V_0 of existing disambiguated links and adding new links. It can be employed as a general purpose iterative semantic analysis tool: in each step k a set of validated targets V_{k-1} is used to re-rank the remaining set C_k of candidates. The links identified and ranked by a densification system allow an arbitrary NLP application to then use those targets that are relevant for the particular task at hand. The task is also related to *wikification* (Cucerzan 2007; Mihalcea and Csomai 2007; Milne and Witten 2008b), a task discussed in more detail in Section 2. Densification can use high-confidence entities identified by wikification and employ them as the starting set V_0, e.g. when processing a document that does not have any annotations.

This semantic representation is an alternative to state-of-the-art approaches for semantic indexing, such as latent semantic indexing (Deerwester *et al.* 1990), probabilistic latent semantic indexing (Hofmann 1999), random indexing (Kanerva, Kristofersson and Holst 2000; Sahlgren 2005) or latent Dirichlet allocation (Blei, Ng and Jordan 2003), which construct semantic spaces based on word co-occurrence patterns in textual documents. Such methods achieve state-of-the-art performance in modern information retrieval engines; however, these models yield abstract conceptual spaces that are difficult to interpret. Using Wikipedia articles as concepts does not have this limitation. Instead of using mathematical criteria to uncover a semantic space, the categories in Wikipedia can be used to create a second-order vector-space that is more natural to interpret. The links between articles allow the use of a generalised vector-space model, i.e. one in which the coordinates are not orthogonal.

The type of semantic analysis proposed in this paper constitutes an advanced type of information extraction, which can be beneficial for a wide range of NLP applications such as information retrieval, text mining, co-reference resolution, automatic summarisation and question answering (QA). It can also help bridge the gap between textual Web documents and Semantic Web resources. This is because the analysis can be performed off-line during indexing/pre-processing, when text can be represented in a semantic space induced by Wikipedia articles. Coordinates of a semantic vector represented in this space correspond to Wikipedia article identifiers, weighted according to the disambiguation confidence score and the number of occurrences. Computing this weighted vector is a form of explicit semantic indexing that enables efficient, higher-level semantic retrieval procedures, enabling queries

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

such as *paragraphs mentioning ⟨Manchester United (the association football team)⟩ and also mentioning a ⟨stadium⟩* to match a headline such as *ManU triumph at Wembley*. The usefulness of this information is demonstrated in Section 5.2 where a QA system capable of answering complex questions is shown.

This paper is structured as follows: Section 2 reviews previous methods for disambiguating named entity mentions and linking text to Wikipedia articles. Most of them aim for balance between precision and recall. To increase the number of links identified and to also include less prominent entities, a novel task – *densification* – is proposed in Section 3. The section also describes an experiment involving human raters carried out to determine whether or not agreement is high enough to warrant the development of a large-scale gold standard annotation. Section 4 presents a two-step densification method that is evaluated in Section 5. In a second experiment, described in Section 5.1, a set of Wikipedia articles are used to automatically estimate the recall of the proposed densification method by measuring how many of the existing wikilinks are also predicted by the system. Section 5.2 describes a third experiment that evaluates the impact the densification system has on EQUAL, a semantic QA system. The paper finishes with conclusions and directions for future work.

2 Previous work

Amongst similar forms of semantic analysis that have been proposed in the past, the most relevant approach is Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch 2007). In ESA, the text of Wikipedia articles is used to create an inverted index: for each word w, the ranked list of documents it appears in is built using $tf \cdot idf$ weighting: $docs(w) = \{(d_w^i, r_i) | i = \overline{1..n_w}\}$, where n_w is the number of documents containing the word w and r_i is the rank assigned to document d_w^i. Given an input text fragment, the system combines the vectors associated with each of its words, yielding an aggregated vector in the semantic space whose coordinates correspond to Wikipedia articles. The authors demonstrate that this space can be used to measure semantic similarity between two texts, even if their lexical overlap is low. ESA is similar to latent semantic indexing approaches in that each word has an association score with every topic (or component of the semantic space), and this score is used to determine how relevant a topic is to a text. The difference is that the components of the ESA semantic space are explicitly determined by a subset of Wikipedia articles, whereas the abstract concepts in the latent spaces cannot be easily interpreted by humans, making them less attractive to some NLP applications, such as information extraction and QA, where to generate explanations or proofs is beneficial for users.

Identifying mentions of Wikipedia concepts in texts, called wikification, is an active research area made popular by Wikify! (Mihalcea and Csomai 2007) and WikipediaMiner (Milne and Witten 2008b). The potential of Wikipedia as a named entity knowledge base was also studied (Bunescu and Pasca 2006; Cucerzan 2007). In addition, more recent tools such as *DBpedia Spotlight* (Mendes *et al.* 2011) as

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

well as commercial systems such as OpenCalais[2] and AlchemyAPI[3] have also been developed. Wikification is a special case of analysis meant to replicate the current linking style of Wikipedia articles, when editors agree on the most relevant articles to be linked, to decrease clutter and facilitate readability for readers rather than pursue comprehensive linking.

There are three main challenges for this type of named entity disambiguation task: determining possible mentions in the input text, determining the best candidate target for each mention individually and selecting a subset of the links that have a high joint confidence score. The last step is necessary because Wikipedia is incomplete and the correct sense might not be present. In the news collection used in the Entity Linking task at the Text Analysis Conference,[4] 57% of named entity mentions did not have a corresponding Wikipedia article. Depending on the nature of the relatedness measure used to rank the targets, there are two main approaches for wikification. The first uses a semantic relatedness measure based on the Wikipedia link graph, while the second uses word-based similarity between text snippets.

A system using the first approach is WikipediaMiner (Milne and Witten 2008b). It considers as a possible mention any word n-gram that has been used as the anchor text of at least one link in Wikipedia. Those mentions that have exactly one possible target make the set of unambiguous targets that is used as a reference context to determine the relatedness to each possible target of the remaining ambiguous mentions. Using this relatedness score, the most likely target article is selected for each mention. In the second step, the system decides which of the targets deemed plausible should be linked based on their relatedness to each other as well as to the reference context. A decision tree model is used to predict the confidence of each link, and only the targets exceeding a fixed threshold are selected. The main contribution of WikipediaMiner is the semantic relatedness measure between two articles a and b, based on the two sets of articles linked to them A and B. The formula is derived from the Normalised Google Distance (Cilibrasi and Vitányi 2007), when using co-occurrence of wikilinks rather than co-occurrence of terms. Formally, the measure is (Milne and Witten 2008a):

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

where $|W|$ is the total number of articles in Wikipedia. The system achieves 97% accuracy in the first step (determining the most valid target of an existing link), exceeding the most frequent target baseline of 90%. While the classifier achieves excellent performance, this result is measured only for existing wikilinks, those mentions marked by Wikipedia editors due to their relevance. In real-life use, the system needs to first determine the mentions themselves, which increases the complexity of the decision space with a negative impact on disambiguation accuracy. The selection criterion in the second step is biased towards prominent

[2] http://www.opencalais.com
[3] http://www.alchemyapi.com/
[4] http://www.nist.gov/tac/about/index.html

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

entities, favouring a nucleus of high confidence and unambiguous targets. The more challenging the disambiguation, the less likely it ranks high enough to pass the selection threshold. This is an effective way to eliminate errors made during the disambiguation step, but only because recall has less importance for wikification evaluation. To further enrich the interlinking in Wikipedia, it is necessary to employ more advanced measures that are able to also link less prominent articles.

A more recent system, TSDW (Li, Sun and Datta 2013), also builds a context from unambiguous mentions and employs a link-based relatedness measure to weigh the importance of context topics and to rank the relatedness of candidate topics to the context. Unlike WikipediaMiner, the system uses two stages for disambiguation: in the first stage, the candidates that have a high confidence are selected and added to the set of seed topics; in the second stage, the remainder of ambiguous mentions are re-disambiguated using the expanded set of seeds. The confidence measure used is the relative difference between the first two ranked candidate topics. By varying the confidence threshold, and therefore the number of candidates selected in the first stage, the system can be biased towards effectiveness or (towards) efficacy. The system achieved high accuracy (0.94), outperforming WikipediaMiner (0.86) on a set of 200 Wikipedia articles. It is unclear whether the impact on accuracy achieved by WikipediaMiner, a drop of 0.11 compared to results reported by Milne and Witten (2008b), is due to the use of a more recent version of Wikipedia or (due) to some details of the experimental settings. The approach is similar to that proposed in this paper because an iterative disambiguation process is employed with new topics being added to the set of seeds; however, our approach has a more complex disambiguation process and aims for a more complete set of links.

A system corresponding to the second approach employing textual similarity was proposed by Bunescu and Pasca (2006). Using the cosine similarity between a mention context and the text of each possible target article yields average results (55%). Applying a support vector machine (SVM) classifier with a taxonomy kernel leads to a 20% error reduction. This kernel exploits the correlations between words and Wikipedia categories, e.g. *conducted* is associated with *Category:Composers* significantly more than with *Category:Wrestlers*. Performance is less impressive than more recent models, but results are difficult to compare because a different data set is used and the most frequent target baseline is not reported. The main limitation of their work is that they only address people mentions that represent a small variation of the title of the corresponding article.

Cucerzan (2007) proposes a large-scale disambiguation method based on Wikipedia data that can only be used for named entities, where mentions are identified by an existing named entity recogniser. A vector representation for each article is used to disambiguate between the candidate entities. The coordinates of this context vector are the wikilinks collected from the first paragraph of an article together with its categories. Disambiguation is seen as an assignment problem: the best solution is found by maximising the agreement between the context vectors extracted from Wikipedia and the context vector extracted from the document. When applied to Wikipedia articles, the system achieves 88.3% accuracy, two points above the most frequent target baseline. This is due to the fact that the majority of

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

named entity mentions are not ambiguous, i.e. they only have one possible target. When applied on a newswire corpus, the system achieves 91.4% accuracy, versus a 51.7% baseline.

Mendes *et al.* (2011) describe DBpedia Spotlight, a supervised memory-based learning approach for disambiguating entities. A dictionary of references is built from the entire contents of Wikipedia: for each article an associated meta-document is built by concatenating the textual snippets surrounding each of its in-links. These meta-documents are indexed by an IR engine. To disambiguate plain text, a query is built for each plausible n-gram and sent to the IR engine, and the highest-ranked article retrieved is selected as the best link. In essence, the disambiguated target is the nearest neighbour in a vector-space representation using cosine similarity between *tf·idf* weighted word-vectors. To evaluate the disambiguation performance, 155,000 wikilinks with ambiguous surface form are set aside for testing, while around 69 million links are used to build the system. A random target baseline only achieves 17.77% accuracy, while the most frequent target baseline achieves 55.12%. The best results were achieved by a regression model that combines the target frequency observed in Wikipedia, i.e. the prior probability of a target, with the cosine similarity score: 80.52%.

thewikimachine[5] uses more than one million kernel-based classifier models, each trained to disambiguate an individual word (Bryl *et al.* 2010). Training data for this supervised WSD task is derived from the textual snippets surrounding wikilinks. To allow the use of a domain kernel exploiting linguistic knowledge, the system uses a mapping of Wikipedia articles to WordNet synsets. The performance of the system is evaluated on the ACE05-WIKI Extension (Bentivogli *et al.* 2010), a data set that extends the English Automatic Content Extraction[6] (ACE) 2005 data set with ground-truth links to Wikipedia. The 599 documents in this collection contain a total of 29,300 named entity mentions manually linked to the Wikipedia articles they refer to. The system achieves an F_1 score of 0.715, compared to 0.587 obtained by WikipediaMiner on the same data set.

In the context of Semantic Web technologies, an area of interest is linking entity mentions in raw text to corresponding entries in linked data sets, a task typically focused on disambiguating names of persons, companies or places. In the case of DBpedia, combining a named entity recogniser with a simple lookup in the large data set compiled from `foaf:name` and `rdf:label` properties (Damljanovic and Bontcheva 2012) yields high recall (0.81) but low precision (0.14). Damljanovic and Bontcheva (2012) propose a simple disambiguation algorithm that combines three similarity scores to rank ambiguous entries: the edit distance between the ambiguous mention and the name of a candidate entry, the cosine similarity between the document text and the DBpedia abstract of the entry (represented in a semantic space using random indexing) and the number of entities from the document that are linked to the candidate entry by DBpedia relations. The algorithm proved effective

[5] `http://thewikimachine.fbk.eu/`
[6] `http://www.itl.nist.gov/iad/mig/tests/ace/ace05/index.html`

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

in eliminating irrelevant candidate entries, achieving much better precision (0.84) with little impact on recall (0.80).

Commercial systems such as OpenCalais,[7] Zemanta API[8] or Alchemy API[9] also allow linking arbitrary text to Wikipedia articles, for example for automatic semantic tagging of newswire articles or blog posts. Such systems are primarily focused on generic named entities (such as person, organisation, location and others). In addition, they emphasise precision and do not exploit existing links. For these reasons, commercial systems are ill-suited as an off-the-shelf solution for densification and are not particularly relevant for this research. However, they can provide candidate links for plain text documents, which could be used as input for densification.

An active research area related to wikification is entity linking, also referred to as record linkage or entity resolution, the task of determining the identity of entities mentioned in a document collection and linking them to their corresponding entries in a reference knowledge base (KB). A popular KB used for entity linking on open domain texts is Wikipedia. Text Analysis Conference (TAC) is the main evaluation forum for the entity linking task. A document collection is used that combines newswire articles and posts from blogs, newsgroups and discussion forums. Given a query that consists of a document and a specified name mention, the task is to determine the correct entry in the reference KB for that mention, or indicating that such an entry is not present in the KB (a NIL answer). The type of named entities considered are: PER (person), ORG (organisation) or GPE (geopolitical entity).

The majority of systems divide the task into three parts (Dredze *et al.* 2010): (a) identifying a subset of KB entries that are reasonable candidates for the queried mention; (b) ranking and selecting the best candidate entry; and (c) deciding between returning the best-ranked candidate and returning a NIL answer to indicate a missing entry. In TAC, more than 50% of mentions did not have a corresponding entry in the KB (NIL answers). Approaches to candidate identification were generally based on name comparisons between the mention and KB entries, often using pre-computed dictionaries. A variety of non-exact matching techniques were used, including: alias lists, character n-grams, phonetic matching, acronyms, Wikipedia links or redirects, external Web search and relationship similarity.

Various approaches have been developed for ranking candidates (McNamee 2009). Some systems use information retrieval techniques for retrieving the correct KB entry, e.g. query expansion (Gottipati and Jiang 2011) or generative clustering models for entities in text based on KB entries (Han and Sun 2011). Other approaches employ learning to rank frameworks (Li *et al.* 2009; McNamee *et al.* 2009, 2012).

For deciding whether or not a NIL answer is to be returned, systems either use a threshold value for the linking confidence, or use supervised approaches to train binary classifiers. Since the mentions to be linked are given in the test set,

[7] http://www.opencalais.com/
[8] http://www.zemanta.com/api/
[9] http://www.alchemyapi.com/

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

performance for this task is typically measured in terms of Accuracy, with systems typically reaching performance above 90% in supervised settings.

3 Densification: task definition

The aim of the densification process is to annotate snippets of text with links to relevant articles from Wikipedia, exploiting existing links if available and incrementally adding new links.

In a way, densification is similar to wikification because both tasks disambiguate entity names and terms by linking them to their corresponding Wikipedia articles. However, there are two main differences between the two tasks. Firstly, wikification aims for strict disambiguation and creates links to coreferent articles, while densification is designed to provide links to relevant Wikipedia pages, enabling other tools to further filter these links to suit their needs. Secondly, densification employs existing links as seed meta-data in the disambiguation context; this makes it possible to increase the density of links in an iterative manner and enables integrating with other existing tools, such as named entity disambiguation models.

For example, in the sentence '*Yesterday, the US President Barack Obama announced...*' a wikification system (WikipediaMiner) predicts two links: ⟨US,United States⟩ and ⟨President Barack Obama,Barack Obama⟩. By contrast, in addition to these links, a densification system finds several overlapping mentions that can refer to the same person: '*the US President Barack Obama*', '*the US President*', and '*Barack Obama*'. It also identifies more than one Wikipedia article relevant to the same mention, e.g. '*the US President*' can also have as target the page *President of the United States*, which describes the office, not the incumbent, or *List of Presidents of the United States*. Other pairs of anchor text and target article that contain relevant information are ⟨*president,President*⟩, ⟨*Obama,Obama (surname)*⟩, even if these are not coreferent in a strict sense. Other candidate targets are not relevant, e.g. ⟨*the,English articles*⟩, ⟨*Yesterday,Yesterday*⟩, the Beatles song.

Parallels can also be drawn to word-sense disambiguation (WSD). On the one hand, the information contained in a Wikipedia article is substantial compared to a WordNet synset gloss, or to a dictionary definition. This information can be used to re-rank the likelihood of a target page by measuring relatedness to other targets. The distance between mentions can be used to only consider targets mentioned locally (in the same paragraph) rather than all targets mentioned globally (in the entire document), thereby making the task less daunting. On the other hand, densification is more difficult than WSD because there are vastly more proper names than common nouns and names tend to be more ambiguous.

In Wikipedia, the same anchor text can be used in links to tens of distinct pages, while individual words, such as 'house', can occur in a very large number of distinct anchor texts linking to e.g. novels, albums, films or landmark buildings. In addition, due to its broad coverage, many entities in Wikipedia are homographs (spelled the same), which means that in a text, mentions with an obvious meaning to a human reader can be very ambiguous for computers. For example, the Wikipedia page *The (disambiguation)* lists four exact homographs, while most NLP tools typically

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

consider 'the' to be the English definite article and discard it when analysing text; the article *President (disambiguation)* lists more than 25 pages that are possible candidate targets, while in WordNet, *president* only has six synsets.

The densification process is also difficult due to the fact that in many cases identification of the mention is not a trivial process. The following sentence illustrates some difficulties in automatically identifying mention boundaries in text: *'The US President Barack Obama sent an official letter of apology yesterday.'* The entire span *'The US President Barack Obama'* refers to the person, but both *'The US President'* and *'Barack Obama'* have the same target. The mention *'the US President'* can also have as target the page *President of the United States*, which describes the office, not the current incumbent. Other plausible ⟨*mention,target*⟩ pairs could be: ⟨*president,President*⟩, ⟨*US,United States*⟩, ⟨*Obama,Obama (surname)*⟩ and even ⟨*The,English articles*⟩.

In the remainder of this section, we present an annotation experiment carried out in order to find out to what extent humans can perform densification and what their levels of agreement are.

3.1 Human evaluation

In order to develop densification systems, an annotated resource or some evaluation benchmarking needs to be developed. Annotators could be employed to create an annotated data set manually. This approach can be employed either to create a reference gold standard data set, or to evaluate systems' output. A pilot study was carried out to determine the feasibility of such an approach. A set of paragraphs from Wikipedia articles were extracted and plausible candidate topics were identified. Human annotators were asked to assess the targets of these links using three criteria: *A-correctness* of the proposed link, *B-relatedness* of its contents to the paragraph and *C-type* of the target article:

- **A-correctness** – the proposed target is a plausible disambiguation for the anchor text in this document. This criterion reflects the ability to discard targets that are obviously incorrect using a three-point scale (Table 1a)
- **B-relatedness** – the content of the target article helps in understanding the current document. This criterion distinguishes between very relevant and completely irrelevant targets using a four-point scale (Table 1b)
- **C-type** – the description included in the target page matches the type of the anchor text. This criterion distinguishes five coarse categories (described in Table 1c)

The experiment conducted focused on four paragraphs of text, totalling 94 links to 48 candidate targets. The six annotators were given an explanation of the scoring scheme as well as examples. In the first step, annotators were shown a short text, usually a paragraph, which they had to read. Then, they had to score the three criteria for each candidate link. The main goal was to study the three criteria, the agreement between annotators and the types of links found in the sample data,

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Table 1. *Criteria assessed by human raters when evaluating target pages proposed for a keyword from a given text*

(a) Criterion A: *Correctness* of the proposed target page		
Correct	A-I	the target proposed corresponds to the correct meaning of the keyword in the text, e.g. the same title, a synonym or an alternative name
Possible	A-II	the target is related to the keyword, but less exact, e.g. keyword *expensive* – target *Cost*
Incorrect	A-III	either incorrect disambiguation or erroneous target that does not match the context, e.g. keyword *foreign visitors* – target *Foreign Policy (magazine)*
(b) Criterion B: *Relatedness* of the proposed target page		
Very relevant	B-I	target page is central to the text, e.g. several valid mentions in the text, or a strong relation to the topic; usually entities or specific terminology
Relevant	B-II	target page describes a named entity or a term referred to in the text; it should have at least one valid mention in the text
Somewhat relevant	B-III	target page provides additional, but less important information (e.g. standard terminology, partial entity names, disambiguation pages)
Not relevant	B-IV	targets describing irrelevant entities or generic terms (e.g. years, dates, measurement units, numbers), or targets deemed *Incorrect* under criterion A
(c) Criterion C: *Type* of the proposed target page		
Named entity	C-I	people, organisations, places, events, albums, etc. e.g. *Paris*, *Picasso*, *World War I*
Term	C-II	common terminology, scientific or technical concepts e.g. *public house*, *capital city*, *isomer*, *CPU*
Topic	C-III	a generic topic, e.g. *Geography of Germany*, *Characters of Lost*, *Age of Enlightenment*, *Partition of India*
Disambiguation	C-IV	a listing of pages with similar titles e.g. *Paris (disambiguation)*
Other	C-V	none of the above e.g. list pages (*List of sovereign states*) or dates (*2003*, *21st century*, *March 01*)

and to gain important insights regarding the feasibility of creating a gold standard annotation for this task.

3.1.1 Correctness

Correctness is measured on a nominal scale with three values. Overall, the six annotators marked 79.17% targets as correct, 14.58% as possible and just 6.25% as incorrect. Due to the skewness of the data set, two inter-rater agreement measures are used: simple agreement, and Krippendorf's alpha (*Ka*) (Krippendorff 2004), a coefficient that considers the level of chance agreements, like Cohen's kappa (Cohen

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Table 2. *Agreement for criterion Correctness*

(a) Pairwise simple agreement

	2	3	4	5	6	Annotator average
1	87.50	87.50	83.33	81.25	83.33	84.58
2		87.50	91.67	83.33	83.33	86.67
3			83.33	77.08	87.50	84.58
4				79.17	81.25	83.75
5					72.92	78.75
6						81.67
						83.33

(b) Pairwise Krippendorf alpha (ordinal)

	2	3	4	5	6	Annotator average
1	70.21	70.81	70.24	58.67	45.96	63.18
2		77.74	88.60	67.73	52.84	71.42
3			67.56	55.74	55.83	65.54
4				67.31	44.35	67.61
5					32.10	56.31
6						46.22
						61.71

1960), and generalises a number of inter-rater agreement statistics, such as Scott's Pi (Scott 1955), Fleiss' kappa (Fleiss 1971) and Spearman's rank correlation coefficient rho (Spearman 1904). *Ka* can be used for various types of data, including nominal, ordinal and interval.

Table 2a shows the pairwise agreement considering categorical ratings. The level of agreement is high, but this does not reflect skewness nor the ordinal nature of the criterion: there are only three cases of 'extreme' disagreement when a target is rated *correct* by one rater and *incorrect* by another. Two of these cases involve the same pair of annotators (5 and 6), while in the third case, the rater felt that the link was 'unnecessary' as the target article was the exact page the text was extracted from (i.e. *Paris*).

Table 2b shows pairwise agreement using Krippendorff's alpha coefficient. For more than two annotators, one can use either the average pairwise *Ka* (61.71%) or the overall *Ka* statistic for all annotators (62.30%). Two annotators (5 and 6), who also spent the least amount of time for the task, achieve substantially lower average pairwise agreement than the first four annotators, suggesting they might have been less exact. Although not very high, the agreement level can be considered satisfactory because labels are not equiprobable and inter-annotator agreement measures are affected by prevalence (Sim and Wright 2005).

3.1.2 Relatedness

One of the purposes of this experiment was to determine whether the proposed criterion 'relatedness' corresponds to a natural attribute of the target, which human

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Table 3. *Agreement for criterion Relatedness*

(a) Pairwise simple agreement

	2	3	4	5	6	Annotator average
1	52.08	39.58	35.42	45.83	27.08	40.00
2		43.75	35.42	64.58	33.33	45.83
3			56.25	41.67	47.92	45.83
4				35.42	39.58	40.42
5					37.50	45.00
6						37.08
						42.36

(b) Pairwise ordinal agreement allowing one-scale-point mismatches between raters

	2	3	4	5	6	Annotator average
1	89.58	95.83	95.83	87.50	83.33	90.42
2		87.50	89.58	93.75	81.25	88.33
3			87.50	85.42	83.33	87.92
4				87.50	75.00	87.08
5					89.58	88.75
6						82.50
						87.50

(c) Pairwise Krippendorf alpha (ordinal) agreement

	2	3	4	5	6	Annotator average
1	48.59	48.38	54.94	48.59	37.65	47.63
2		44.05	52.97	66.85	52.54	53.00
3			56.33	45.41	48.29	48.49
4				59.50	46.34	54.01
5					64.41	56.95
6						49.85
						51.66

raters can recognise without special training. The guidelines for rating relatedness were minimal and described the scale and provided several examples, rather than consisting of a strict set of rules. The criterion was measured using a four-point ordinal scale.

As Table 3a shows, the average simple agreement is substantially lower than for the previous criterion. To reflect the ordinal nature of the scale, we can use a modified version of the agreement measure that allows for one-point disagreements between a pair of raters. Table 3b shows that the majority of the pairwise disagreements are differences of one scale point. This means that agreement was relatively low when relatedness ranking is seen as a categorical rating task, but higher agreement levels were noticed when using an ordinal index.

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Table 4. *Number of targets per label for criterion Relatedness*

	Label			
Rater	I	II	III	IV
1	6	30	9	3
2	10	24	7	7
3	12	14	18	4
4	23	5	15	5
5	11	18	10	9
6	10	7	16	15
Average	12.00	16.33	12.50	7.17
Median	10.50	16.00	12.50	6.00

Table 3c shows that, overall, the level of agreement using Krippendorff's alpha coefficient is indeed lower than for the previous criterion, with only half of raters achieving an average pairwise index above 60% (raters 2, 4 and 5). The pair that agrees best (raters 4 and 5) have a moderate agreement index of 66.85% while the average pairwise index is 51.84%. This suggests that there is too much subjectivity when assessing criterion *B-relatedness* on a four-point ordinal scale. Table 4 illustrates this point: rating *B-I* (Very Relevant) has significantly higher prevalence for annotator 4 (23 items, versus a median of ten items); rating *B-II* (Relevant) has high prevalence for annotators 1 and 2 (though they agree on this label less than half the time); while rating *B-IV* is used by rater 6 twice as frequently as the overall average. The disagreements are inconsistent even for pairwise decisions. By mapping the ratings to a smaller scale (either three-point or even two-point), simple agreement increases but *Ka* decreases slightly. This strongly suggests that the degree of subjectivity in the decisions made by the raters is very high. The conclusion is that *relatedness* is a less generic and a more subjective concept than *correctness*: whether a target is deemed related to a text or unrelated depends on the task at hand, as well as on the rater's knowledge of the topic. Some raters suggested 'relevance' is a more familiar criterion (targets that are mentioned more often are more relevant), but this criterion does not consider the actual contents of the target page.

3.1.3 Type

The third criterion was meant to be used as a control signal, as the distinction between named entities, terms and topics seemed a natural intuition. Analysis of this data revealed two notable findings. The first finding concerns the label *C-V (Other)* that is meant to be used for index pages from Wikipedia, such as lists of entities or dates/years. These pages are important for aiding navigation but are not an encyclopaedic article. A high agreement level was expected for this type of page, but this turned out not to be the case. Results show that raters 4 and 5 did not use the whole scale to rate targets, suggesting perhaps an omission in the guidelines. The second finding is that raters had a very low agreement for label *C-III (Topic)*. The

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Table 5. *Agreement for criterion Type*

(a) Pairwise simple agreement

	2	3	4	5	6	Annotator average
1	77.08	66.67	70.83	58.33	68.75	68.33
2		83.33	89.58	72.92	79.17	80.42
3			79.17	72.92	75.00	75.42
4				72.92	70.83	76.67
5					75.00	70.42
6						73.75
						74.17

(b) Pairwise Krippendorf alpha (nominal) agreement

	2	3	4	5	6	Annotator average
1	65.41	47.20	57.30	39.18	51.06	52.03
2		72.28	84.14	58.85	66.04	69.34
3			66.73	57.01	56.88	60.02
4				60.05	54.28	64.50
5					60.61	55.14
6						57.77
						59.80

typical disagreement was for targets such as *Economy*, which could be considered both general terms and topics in human knowledge. Most topics present in the data set are rather generic and are affected by this problem. More specific topics such as *Economy of France*, which are easily distinguishable from terms such as *Gross Domestic Product (GDP)* are less frequent in the data used. To address this issue, the examples given as guideline need to be amended to better illustrate the differences between topics and terms.

When measuring the agreement on one label versus all the others, raters agreed best for label *C-I (Named Entities)* with Krippendorf's alpha coefficient of 79.4%, while for label *C-II (Terms)* the agreement was 64%. The other labels, which were used for significantly fewer targets, achieved lower levels of agreement: 50.3% for labels IV and V (combined), and 4.62% for label *C-III (Topic)* due to its very low prevalence. If labels are mapped on a three-category scheme (named entities, terms and other), the agreement amongst all raters is 65% and amongst the three best raters it is 77.3%. The best pair of annotators (2 and 4) achieve a Krippendorf's alpha coefficient of 84.1% using the original five-category scale. For this pair of raters there is no target they simultaneously label as *C-III (Topic)*, but they only disagree when one of them uses label *C-III* (10% of targets), while all the other targets have perfect agreement. This is a positive result, showing that strong agreement can be achieved, but at the same time it is substantially above the average pairwise agreement. The detailed agreement results for the *Type* criterion can be found in Table 5.

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

3.1.4 Conclusions

Overall, the results of this experiment suggest that the level of agreement is average for criteria *B-relatedness* and *C-type*, and good for criterion *A-correctness*. Some of the disagreement can be explained by the following facts: the guidelines were not comprehensive, there was no specialised training for the student raters and the distribution of the scores was significantly skewed, which explains the large difference between simple agreement and Krippendorff's alpha, which is adjusted for chance.

The experiment was meant to determine the reliability and generality of the first two criteria, i.e. whether they correspond to natural attributes characterising a target. For this reason the guidelines were intentionally rather vague, to allow raters to use their common sense rather than relying solely on a comprehensive set of rules. While the notion of *correctness*, a fuzzy version of 'identity' or 'cross-document co-reference' seems to correspond to this intuition, *relatedness* appears to be a subjective, task-dependent and somewhat artificial concept. This finding casts doubts over the practicality of developing a large-scale, general-purpose gold standard annotation for *relatedness*. Based on these results, voting amongst raters can be used to evaluate the output of densification systems for *correctness* and *type* using a slightly amended set of guidelines. More rigorous training of the raters as well as quality monitoring measures are needed. However, the main finding of this annotation experiment is that this approach cannot be used for creating a large-scale gold standard to develop densification models and evaluate their intrinsic performance in a supervised setting. Regardless of the amount of training or the number of raters employed in a crowd-sourcing experiment, the data set will always be incomplete because it cannot account for entities that do not have a Wikipedia article and because of the way in which mentions can overlap. In light of this, a more pragmatic way to measure the performance of densification systems is via extrinsic evaluation.

4 Densification: a two-step approach

Given a textual document d and a set of n seed topics $S = \{s_i | i = 1 \ldots n\}$, the task is to find a set of topics $T = \{t_j\}$ that are relevant to entities mentioned or referred to in the document, where each topic corresponds to one Wikipedia article. We use the term 'topic' to emphasise that relevant pages do not imply coreferential identity relations. The system has to identify candidate topics and rank them by importance, integrating local information extracted from snippets of d with global information extracted from Wikipedia. The system collects $\langle$*anchor text*,*target page*$\rangle$ pairs from all the wikilinks in Wikipedia. Other alternative names are extracted from redirect pages, disambiguation pages and title variants. Any span ref_k from d, which is a known anchor text (or alternative name), becomes a possible reference. All known targets of a possible reference ref_k are considered candidate topics $ct_{k,l}$. This yields a low precision–high recall list of candidates $C = \{ct_{k,l} | k = 1 \ldots m, l = 1 \ldots r_k\}$, which needs to be filtered.

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

In order to perform this filtering, the system employs a two-step approach: pruning and selection. In the first step, candidate topics are filtered out based on their semantic relatedness to the set of seeds and to each other. For each mention ref_k the topics with low overall confidence are removed. Section 4.1 describes how machine learning is used to achieve this. This step significantly reduces the number of candidates, while maintaining high recall. In the second step, the system determines which of the remaining candidates should be selected also by using the semantic relatedness to other topics, but giving more weight to the local context of the actual topic mentions and dealing with overlapping references. The aim of this step is to increase precision. This is described in Section 4.2.

4.1 Candidate pruning

Candidate pruning is essentially a weaker form of disambiguation. For each mention ref_k candidate topics that are semantically relevant to the document are passed to the next step, while those with low relevance are removed. This step increases precision while maintaining very high levels of recall, to discard weak candidates and to decrease complexity.

4.1.1 Context weighting

The system relies on the relatedness between candidate topics and the reference set of seed topics (the context). The importance of each seed topic s_i is first ranked based on occurrences in the document d as well as frequency information from Wikipedia and average semantic relatedness between seeds. Several ranking methods were tested empirically, as there is no 'gold standard' for ranking the set of seed topics to enable the direct measurement of performance. These methods assign a weight to a seed as follows:

- **simple** – all seeds have equal weight: $w_i = 1$
- ***tf*** – the weight of seed, s_i, is proportional to tf_i, the number of occurrences of its known anchor texts in the document (possible seed mentions): $w_i = \log(1 + tf_i)$
- ***tf·idf*** – considers the number of possible mentions in the document as well as the number of Wikipedia articles that have a link to each seed: $w_i = \log(1 + tf_i) \cdot \log(\frac{W}{inlinks_i})$, where W is the number of articles in Wikipedia, $inlinks_i$ represents the number of articles that link to the topic s_i
- **avg-rel** – the average semantic relatedness to the other seeds, using the measure proposed by Milne and Witten (2008b): $w_i = \frac{1}{N} \sum_{j=1}^{N} rel(s_i, s_j)$
- **avg-rel-*tf·idf*** – combines the previous method with the weights provided by the ***tf·idf*** method: $w_i = \frac{\sum_{j=1}^{N} rel(s_i,s_j) \cdot w_j^{tf \cdot idf}}{\sum_{j=1}^{N} w_j^{tf \cdot idf}}$
- **links-in** – reflects the number of seed topics that have a direct wikilink to this article: $w_i = |S \cap inlinks_i|$
- **links-out** – the number of seed topics that article s_i has a direct wikilink to: $w_i = |S \cap outlinks_i|$

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

- **links-all** – the number of seed topics that article s_i is linked to: $w_i = |S \cap \{outlinks_i \cup inlinks_i\}|$

4.1.2 Candidate relevance

The next step is to rank the importance of each candidate topic $ct_{k,l}$ independently. The candidates whose relevance does not pass a threshold value are discarded, thus pruning the decision space. The threshold is a parameter that controls the trade-off between *precision* and *recall*. To solve this problem, a pointwise learning-to-rank approach is employed.

A machine learning data set is created using features that characterise the context (number of seeds, average weights), the candidate topic (prior link probability, number of references, number of in- and out-links) and the compatibility/relatedness between the candidate topic and the set of seeds. To create training data, a random set of Wikipedia articles is used. For each article, a fraction of its wikilinks are considered as seeds, while the remainder make the set of valid targets. Only a subset of all the ambiguous references are used, namely those mentions $ref_{k'}$ that have at least one candidate topic amongst the target articles. The resulting data is split into training and development sets, to avoid problems that can arise when using cross-validation alone, such as duplicate instances with copies in more than one cross-validation fold. When applied to real-world documents, while the seed set must be supplied as input, the set of valid targets is not known and the model must make predictions for all ambiguous mentions ref_k. Without a gold standard created specifically for densification, this bias induced by sampling the training data set is unavoidable. By varying the amount of seed data, we can gain insights into the performance of the model.

4.1.3 Experiments

A random set of 100 articles were selected for training the models, and another random set of 100 articles were used for testing. Point-wise ranking is employed: for each candidate the model produces a confidence score. Then, for each reference a simple criterion is applied to select candidates, e.g. the candidates that are either in the top three highest-ranked candidates per reference or have a high confidence, above a threshold value.

The seeds are weighted with the methods mentioned in the previous step, while the features describing the topic are directly extracted from Wikipedia statistics. The most important features are those reflecting the semantic association between a candidate topic and the set of seeds. Two base measures are used: WikipediaMiner's semantic relatedness measure, which is based on the Normalised Google Distance (Cilibrasi and Vitányi 2007) between the sets of links of two articles, and a measure that accounts for direct mentions/links between the articles, information that is not used by the previous measure. The base measure needs to be aggregated to account for the fact that the context has several seeds, each mention has several candidates and each candidate has several possible mentions. For direct links, we use two

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Table 6. χ^2 *ranking of the candidate pruning features used to measure the relatedness between a candidate target and the set of seed topics*

Average χ^2 merit and variation		Average rank	Relatedness measure	Aggregation method	Seed topics' weighting
7091.067	±30.662	1	WM	max	*tf·idf*
6826.216	±22.576	2	WM	top10	*tf·idf*
5750.507	±26.079	3	WM	avg	avg-rel
5654.750	±19.512	4	WM	top10	*tf*
5526.763	±13.919	5	WM	avg	simple
5467.932	±16.275	6	WM	avg	avg-rel
5102.377	±46.585	7	WM	max	*tf*
4850.794	±25.915	8	WM	avg	links-out
4687.179	±20.891	9	WM	avg	link-all
4643.823	±30.256	10	WM	top10	avg-rel
4550.655	±28.174	11	WM	avg	*tf·idf*
4258.057	±46.608	12.7	WM	max	avg-rel
4237.392	±25.958	12.8	links	count	links-in
4213.878	±27.175	13.5	WM	avg	*tf*
4149.599	±26.924	15	WM	avg	links-in
4054.404	±20.434	16	WM	top10	simple
3010.643	±18.823	17.2	WM	top10	avg-rel
2997.218	±23.362	17.8	WM	max	simple
2816.212	±36.212	19	links	count	links-all
2433.655	±32.184	20.5	links	islink	links-all
2434.678	±11.281	20.5	WM	top10	links-out
2279.353	±20.949	22.1	WM	max	avg-rel
2214.805	±24.357	22.9	links	count	links-out
2023.238	± 9.356	24	WM	top10	links-all
1907.707	±13.352	25	WM	max	links-in
1838.768	± 7.598	26	WM	top10	links-in
1676.377	±13.025	27	links	islink	links-in
1519.758	±16.815	28	WM	max	links-out
1071.573	± 9.190	29	WM	max	links-all
868.720	±15.587	30	links	islink	links-out

aggregation methods: link count (*count*) and has-at-least-a-link indicator (*islink*). For semantic relatedness, we use: average (*avg*), maximum (*max*) and average of the top ten nearest seed topics (*top10*). The different seed topic weighting methods are combined with different relatedness functions with these aggregation methods. The resulting relatedness scores are used as features to train supervised models. Table 6 shows the top-ranked features using the χ^2 statistic.

4.1.4 Results

For comparison, the open-source WikipediaMiner (WM) package was used to train and test a disambiguation model on the same set of articles (see Table 7). The WM model performs slightly worse than the original results reported on a different sample of articles from an older version of Wikipedia. This suggests that the problem is

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Table 7. *Candidate pruning performance*

Model	Accuracy	Error rate	Kappa
WM-disambig	91.33%	8.67%	0.7551
Logistic regression	95.37%	4.63%	0.7773

Table 8. *Trade-off between recall and precision for candidate pruning*

Threshold θ	R	P	$F_{0.5}$	F_1	F_2
0.40	73%	87%	84%	79%	75%
0.30	76%	85%	83%	80%	78%
0.05	97%	49%	54%	65%	81%

more difficult in the newer version of Wikipedia, due to the increase in size, number of pages/links and article length. The main direct cause could be that, on average, the frequency of the most common sense has decreased. The WM model relies on the prior probability of a target given the anchor text, which amounts to a strong bias towards the most frequent target baseline. According to Cucerzan (2007), this baseline yields better results on text from Wikipedia than it does on newswire text.

Overall, the features obtained using the WM semantic relatedness measure had higher predictive power than those using the link-based measure. None of the aggregation methods consistently outrank the others. As expected, there is a high positive correlation between features resulting from applying different aggregation methods to the same base relatedness measure.

A logistic regression trained using all the numeric features described above outperforms the state of the art, halving the error rate. Logistic regression is suitable for our binary data set, as the confidence value output by the algorithm is actually the probability that the instance is a valid topic. This means that sorting topics based on this confidence score yields a robust ranking, which is desirable as it allows the use of a threshold value to favour recall over precision. Table 8 illustrates that by varying the hyper-parameter θ the system can be biased towards achieving high recall (few false negatives). This model reaches impressive recall levels (97%) while maintaining reasonable precision (50%). This result is in line with this stage's bias towards recall. In this data set, on average, there are ten candidates for each mention, one of which is considered valid. Using the regression model discards, on average, eight out of ten candidates, but it almost never discards the target of the original link. Table 8 shows the threshold values that yield the best precision, recall or F_1 score.

4.2 Candidate selection

In the second step, the focus shifts from the global context, the overall relatedness to all seeds, to the local context of each mention. For each mention, a score is computed factoring in the relatedness to seeds and to candidates with possible

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

mentions nearby. For each nearby mention, only the most related candidate is considered. The distance between mentions, measured in words/tokens, is used to weight the relatedness score. The length of a mention is also used as a feature, because longer anchor texts tend to be less ambiguous. The weighted relatedness score of candidate topic t_k of mention m to the topic t_i of a nearby mention m_i is: $\frac{rel(t_i,t_k)\cdot\log(length(mi))}{\log(dist(mi,m))}$, where $rel(t_i, t_k)$ is the WikipediaMiner relatedness score. Like in the previous experiment two aggregation functions are used, *max* and *avg*, and the aggregation is done separately for seeds and for ambiguous mentions, thus creating four numeric features. Short mentions overlapped by long mentions should be linked to one of the targets of the surrounding mention, and other possible target topics should be discarded, therefore a binary feature is used to indicate this. These numeric features can be used to train a classifier or a regression model.

Unfortunately, as its markup is incomplete, information extracted from Wikipedia is no longer sufficient to create training data automatically for this step. Without validation data to measure the performance of a particular model, an extrinsic validation methodology was used to select the parameters of a relevance estimation function, a linear combination of the numeric features. For each mention, the system ranks all candidate topics, then selects those at the top of this list. The overall number of topics selected can be expressed relative to the total number of candidates or to the number of seeds.

To determine a good set of weights for the ranking function, a supervised document classification task was employed using a subset of the 20-newsgroups data set[10] (Lang 1995) consisting of 100 files for each newsgroup. After removing headers and e-mail addresses, the resulting 2,000 files average 272.93 words. Although this data set was used for benchmarking classification systems, and many other evaluation data sets exist, the experiment was not meant to compare classification performance achieved via densification with state-of-the-art results, but instead it was the extrinsic methodology employed for parameter selection. Unlike the usual 60-40 split between training and testing (or the 90-10 split used in ten-fold cross validation), only 10% of the documents are used for training and the remaining 90% of data is used for testing, because such a split forces a document classifier to generalise, i.e. to rely more on semantic relatedness rather than lexical overlap. The intuition is that the best performance in this classification task is achieved by choosing a good relevance weighting function. All documents are first wikified using WM to select the five most important topics per document as seeds. Documents are then represented as feature vectors, each coordinate corresponding to a Wikipedia article weighted according to the relevance score computed by the densification system. Different relevance estimation functions are used with different weights. As a baseline, null-relevance densification is used: it passes through the input seeds, but does not add any new topics. This is equivalent to the wikification system alone. A grid search was used to select the best performing set of weights for second

[10] `http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html`

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

step. The evaluation of the entire densification system is presented in the next section.

5 Evaluation

The main reason why wikification is popular is the availability of large amounts of training and testing data directly from Wikipedia. The existing approaches give much more weight to predicting precise prominent targets, rather than to completeness. Densification aims to find new links in addition to the existing ones, therefore both precision and recall need to be calculated to measure performance. This can be achieved by manually creating a reference gold standard annotation, which can be costly and, as shown in Section 3.1, is a difficult process. Therefore, it is necessary to also consider alternative ways to estimate performance.

5.1 Wikipedia markup evaluation

This section evaluates the performance of densification by automatically estimating recall. In this experiment, the text of Wikipedia articles is used together with a random sample of the existing links that form the set of seed topics. The set of remaining links, deemed relevant by Wikipedia editors, is considered a biased sample of important links the system should find and is used as a proxy to estimate recall – a subset of valid targets. For each document, the recall score is averaged over ten random splits. The main advantage of this method is that it can be applied on a large scale as it does not require manual data. The disadvantage is that precision cannot be estimated automatically, but needs to be estimated by humans, using a method similar to that described in Section 3.1. Off-the-shelf wikification systems should perform well in this scenario, as only the 'prominent' links are considered for evaluation. Different seed ratios were used to study the impact of input set size.

To better understand the performance of the proposed densification model, several off-the-shelf tools were also tested on the same data set. The tools were briefly described in Section 2. A summary of their characteristics is outlined in Table 9. Almost all tools identify both entities and terms, though not all directly create links to Wikipedia or to the Linked Open Data cloud. Four out of the seven systems provide descriptions of their approach. Two of them use a link-based measure to rank and disambiguate targets, and two systems use a word-based approach. An analysis of popular named entity extractors for linked data is given by Rizzo and Troncy (2011).

Performance is measured in terms of Precision (P) and Recall (R) with respect to the subset of valid targets extracted from the original wiki markup. As argued before, Precision cannot be calculated for densification due to incomplete annotations. Although the results are not directly comparable, they provide a context to better understand and interpret densification performance. Table 10 presents the performance achieved by the different wikification tools using different seed ratios. Only WikipediaMiner has a confidence threshold parameter that can be used to control the bias between precision and recall. The recommended value was used in

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Table 9. *Wikification tools*

	Alchemy API (aapi)	DBpedia Spotlight (dbs)	Open Calais (oc)	thewiki machine (twm)	Wikipedia Miner (wm)	Zemanta API (zem)
Link targets						
• entities	✓	✓	✓	×	✓	✓
• terms	✓	✓	×	✓	✓	✓
Similarity						
• word-based	?	✓	?	✓	×	?
• link-based	?	×	?	×	✓	?
Open source	×	✓	×	×	✓	×
Fair-use	✓	✓	✓	✓	✓	✓
Quota	✓	✓	✓	✓	×	✓
Subscription	×	×	×	×	×	×

this experiment. For WikipediaMiner two models were used, *wm1* was trained on a version of Wikipedia from 2007 and *wm2* used data from 2009.[11]

The first thing that is noticeable is that each system has consistent performance, regardless of the seed rate. This is expected, as these tools do not employ the reference set of seed entities to aid disambiguation. As the seed rate is increased, the remaining set of valid targets is smaller, but the output of each tool does not change. Recall is more stable relative to the seed rate; while small variations do occur, these are more random and of small magnitude. A smaller set of valid targets means a smaller denominator in the recall formula. The random selection of seeds further impacts the actual values, therefore results are averaged over ten runs.

The most curious behaviour is displayed by *thewikimachine* tool, which achieves very poor performance, much lower than previously reported results (Bryl *et al.* 2010), albeit in a different experiment. This is likely because it tries to disambiguate simple terms using specially trained SVM classifiers, rather than focus on encyclopaedic named entities, as seems to be the case with all other tools. The data can be used to rank wikification performance, e.g. by examining the results for the minimum seed rate. Overall, WikipediaMiner does seem to reach higher Recall values. The Zemanta API seems to edge ahead of WikipediaMiner in terms of Precision, but the difference is not statistically significant. This experiment is not meant to benchmark and rank the performance of these tools, but rather to provide insights into understanding performance when only using a subset of the valid targets.

A full comparison between the results obtained by wikification tools and the densification method proposed here is not possible because they address different tasks. However, for the sake of understanding the results better, the results of the densification method are estimated in a similar framework as wikification. Table 11 shows the results achieved by the system as well as those achieved by Wikipedia

[11] Available at `http://sourceforge.net/projects/wikipedia-miner/files/data/`

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Table 10. *Performance achieved by wikification tools using different seed ratios*

(a) Macro-averaged Recall (and standard deviation)

System	sr = 0.1		sr = 0.3		sr = 0.5		sr = 0.7	
aapi	29.97	(15.95)	30.03	(16.11)	30.25	(16.16)	31.54	(17.05)
dbs	48.68	(16.70)	48.48	(16.86)	48.20	(16.26)	48.91	(18.37)
oc	25.44	(14.39)	25.34	(14.22)	25.13	(14.77)	27.64	(14.94)
twm	8.20	(17.24)	8.89	(17.95)	9.85	(18.15)	12.49	(21.37)
wm_1	61.80	(18.16)	61.88	(17.88)	61.13	(17.89)	61.54	(20.68)
wm_2	61.81	(18.62)	61.89	(18.26)	61.19	(18.41)	61.86	(21.20)
zem	57.36	(17.40)	57.53	(17.60)	57.50	(17.09)	57.19	(18.88)

(b) Macro-averaged Precision (and standard deviation)

System	sr = 0.1		sr = 0.3		sr = 0.5		sr = 0.7	
aapi	52.86	(15.52)	47.41	(15.31)	40.16	(14.67)	30.13	(13.47)
dbs	32.89	(11.92)	27.92	(11.28)	22.02	(10.03)	14.95	(8.18)
oc	51.93	(15.42)	45.91	(16.25)	38.20	(15.96)	29.28	(14.55)
twm	7.97	(8.17)	6.90	(7.35)	5.55	(6.70)	4.28	(4.83)
wm_1	57.03	(15.00)	51.50	(15.11)	43.59	(15.17)	32.80	(14.42)
wm_2	46.34	(13.94)	40.79	(13.70)	33.31	(13.29)	23.94	(11.92)
zem	59.85	(11.90)	54.14	(12.11)	46.29	(12.71)	34.69	(11.89)

(c) Macro-averaged F_1 (and standard deviation)

System	sr = 0.1		sr = 0.3		sr = 0.5		sr = 0.7	
aapi	35.70	(15.04)	34.12	(14.65)	31.81	(13.44)	28.27	(12.73)
dbs	37.12	(10.83)	33.40	(10.58)	28.29	(9.32)	21.48	(8.34)
oc	31.73	(13.94)	30.24	(13.31)	27.68	(12.53)	26.05	(11.11)
twm	6.66	(9.34)	6.47	(8.65)	6.04	(8.09)	5.45	(6.35)
wm_1	57.17	(14.04)	54.00	(13.68)	48.60	(13.61)	40.58	(14.55)
wm_2	50.82	(13.11)	47.00	(12.51)	40.96	(12.09)	32.59	(12.32)
zem	56.07	(11.95)	53.14	(11.36)	48.61	(10.72)	40.57	(11.20)

Table 11. *Densification: results for Wikipedia markup evaluation (seed rate 0.5)*

System	Recall	Precision	(Precision upper bound)
WikipediaMiner v.1	46.60%	51.07%	n/a
Densification 1:1	58.73%	57.26%	100.00%
Densification 2:1	71.27%	24.42%	50.00%
Densification 3:1	83.23%	14.51%	33.33%
Densification 4:1	87.13%	9.62%	25.00%

Miner v.1, which performed best amongst wikification tools. In this evaluation setting a seed rate of 0.5 was employed: the number of links used as seeds is equal to the number of reference targets used to estimate recall. Four different threshold values were used to specify the ratio between the number of topics predicted by

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

densification and the number of reference targets sought. The results show how recall increases when more densified links are considered, but this also reduces the upper bound of precision because, although many of these topics are indeed relevant, they are not present in the set of reference targets and are considered false positives. In terms of recall, the densification method performs very well: the average recall is 71.27% when twice as many topics are considered than the number of reference links (Densification 2:1). This type of evaluation is best suited to the comparison of densification systems, but does not fully show the usefulness of densification. In the next section, an extrinsic evaluation is carried out.

5.2 Densification and question answering

Open-domain QA is one of the NLP applications that can benefit from densification tools. QA systems enable users to search for specific pieces of information in large collections of texts by providing exact answers to natural language questions (Harabagiu and Moldovan 2003).

QA systems are typically built around an information retrieval engine that identifies paragraphs of text similar to the keywords used in the question and then extracts a list of plausible answers (Hirschman and Gaizauskas 2001).

In the past decade, open-domain QA research has been dominated by evaluation forums such as TREC (Voorhees 2001) and CLEF[12], where shallow techniques relying on information redundancy have achieved very good performance. But typically QA systems target a specific taxonomy of questions (Hovy *et al.* 2001; Li and Roth 2002). For each type of question, specialised queries are used to retrieve paragraphs from a document collection (Harabagiu, Maiorano and Pasca 2003). Candidate answers are extracted based on various criteria such as similarity to the question (Moldovan *et al.* 2007), or semantic compatibility (Schlaefer *et al.* 2007). These are then ranked (Chu-Carroll *et al.* 2003; Jijkoun *et al.* 2007) and filtered (Moldovan *et al.* 2007).

5.2.1 Encyclopaedic question answering

Relying on information redundancy is less suitable for more complex questions, whose answers are not phrased explicitly in one snippet of text and instead require the system to deduce them by combining evidence from several sources. Examples of such questions are: *Which Brazilian football players play in clubs in the Iberian Peninsula?* or *Which Portuguese rivers flow through cities with more than 150,000 inhabitants?* These are *encyclopaedic questions* – open list questions that explicitly ask for a specific type of named entity and usually contain additional conditions or constraints (Dornescu 2012). They are challenging because instead of having a single correct answer, an entire set of named entities has to be identified and validated individually. To address this type of question, QA systems need to go beyond keyword-based paragraph retrieval, and densification tools can provide metadata

[12] `http://clef.isti.cnr.it/`

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

GP3: Portuguese rivers that flow through cities with more than 150000 inhabitants

Douro, Mondego

GP9: Composers of Renaissance music born in Germany

Arnolt Schlick, Christoph Demantius, Conrad Paumann (12 more...)

GP6: Which Australian mountains are higher than 2000m?

Mount Kosciuszko, Mawson Peak, Mount Jagungal, Mount Townsend

(a) GikiP 2008 topics

GC09-01: List the Italian places which Ernest Hemingway visited during his life.

GC09-04: Name Romanian poets who published volumes with ballads until 1941.

GC09-09: Name places where Goethe fell in love.

GC09-11: What Belgians won the Ronde van Vlaanderen exactly twice?

GC09-29: Places above the Arctic circle with a population larger than 100,000 people

GC09-37: Which Norwegian musicians were convicted for burning churches?

(b) GikiCLEF 2009 topics

Fig. 1. Sample of GikiP and GikiCLEF topics.

enabling more complex forms of semantic information retrieval than otherwise possible.

Two QA evaluation campaigns focusing on challenging list questions were GikiP (Santos *et al.* 2009) and GikiCLEF (Santos and Cabral 2009). GikiP was a pilot QA task that took place in CLEF 2008 and combined QA and geographic IR. The requests for information consisted of natural language topics, phrased either as a question or as an imperative retrieval directive, which had some sort of geographical constraint. The answer was supposed to be a set of results – Wikipedia article titles describing the entities that correspond to the information need that is expressed in the topic (see Figure 1a). The GikiP pilot consisted of 23 topics: 8 for development and 15 for testing, all available in three languages: English, German and Portuguese. GikiCLEF was an evaluation task organised at CLEF 2009 and was a successor of GikiP. As with its predecessor, it also consisted of geographically challenging information requests (see Figure 1b). Systems had to search the Wikipedia collections and return answers as lists of document titles. GikiCLEF emphasised the multi-lingual component using ten language versions of Wikipedia: Bulgarian, Dutch, English, German, Italian, Norwegian (both Bokmål and Nynorsk), Portuguese, Romanian and Spanish. The evaluation campaign used 24 development topics and 50 test topics, all translated into the ten languages of the competition.

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

5.2.2 EQUAL

EQUAL is a QA system representing a novel, semantic approach that uses Wikipedia as a source of world knowledge and employs simple forms of open-domain inference to answer encyclopaedic questions (Dornescu 2010). To address natural language ambiguity, EQUAL generates possible semantic question interpretations and attempts to answer each of them separately. A question interpretation consists of high-level semantic constraints corresponding to the conditions identified in an encyclopaedic question. To find answers, EQUAL uses the Wikipedia link graph to search for relevant information and to test whether each candidate answer satisfies these semantic constraints, considering evidence from multiple documents and/or sources.

In EQUAL, the question analyser examines the structure of a GikiCLEF question using linguistic information and identifies a set of possible constituents that should each correspond to an atomic condition or constraint. The question interpreter determines the compatible semantic constraint types for each constituent (sub-filter). Due to ambiguities in the context of the semantic constraints that the system 'understands' and the way the relevant information is present in Wikipedia and any associated data sources, several semantic interpretations are usually generated that are then answered separately.

All data used by the system is represented as a huge graph: a node denotes a single named entity associated with a Wikipedia article; the edges between nodes correspond to wikilinks between articles and represent untyped relations between named entities. The nodes aggregate various data associated with the named entity, including the text of the Wikipedia article and its categories, data extracted from infoboxes and information from semi-structured datases such as DBpedia and Yago. To find and validate answers for a question interpretation, EQUAL first identifies a set of candidate named entities and then explores the Wikipedia-induced graph to verify which of them satisfy the semantic constraints. Validating a constraint typically involves searching for evidence either in the data stored within the particular node, or, when the constraint involves relations to other entities, in the data contained by the immediate neighbours. This method of searching for evidence achieves very good precision but assumes that all necessary edges are present in the graph. This is a strong assumption because Wikipedia is incomplete and is intended for human readers and only the most relevant entities are linked.

In the CLEF competitions, EQUAL demonstrated the highest performance amongst all submissions, regardless of whether they were automatic or they had human involvement and it also retrieved more correct answers than any other participant. An important performance bottleneck for the system was the limited types of semantic constraints: for some questions, although phrased in a simple form, the system could only produce an approximate question interpretation. Despite having few, generic components, the interpreter managed to return answers for half the questions. As new types of semantic constraints are added and more constraint validators are included in the system, increasingly large amounts of data are necessary to accurately rank the most likely interpretations for the questions involved. This requires a coordinated community effort, especially due to the high

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

degree of interdisciplinary knowledge involved, and also due to the amount of data and resources necessary.

5.2.3 *Densification in* EQUAL

Densification can be used to address the main performance bottleneck that affects the questions for which EQUAL found at least one compatible interpretation. In many such cases, the QA system did not find all of the answers when exploring the Wikipedia link graph, because the constraint verifiers employed depended on wikilinks. However, these are meant for human readers, and are used accordingly. In cases where some of the correct answers were overlooked, this was usually due to 'missing links'.

The evaluation is performed on the GikiCLEF 2009 data set with exactly the same question interpretations generated by the system. This process can provide answers to two questions: (a) does the Wikipedia link graph enhanced by densification improve recall? (b) how does it affect precision? Results are compared to those achieved by the original system.

EQUAL primarily uses two graph-exploration steps: (a) visiting out-links (the entities mentioned in an article), and (b) visiting in-links (the articles mentioning a named entity). Too much computational effort is necessary to densify the entire Wikipedia beforehand, and therefore only those articles that are relevant to each question were processed, on-demand. It is straightforward to densify an article before collecting its out-links. To augment the set of in-links for a given entity e, an inverted index built by EQUAL using Lucene (Hatcher and Gospodnetic 2004) is used to retrieve candidate documents as follows: for each known anchor text a all articles that contain a and do not contain a link to e are retrieved, excluding the documents that already have a link to known targets t_a of anchor text a. These are filtered out because densification would use this target as a seed.

In its original GikiCLEF submission, EQUAL returned 79 correct and 59 incorrect answers on the English data set out of an answer pool containing 118 validated answers. This achieves a micro-averaged precision of 57.24% and a micro-averaged recall of 66.94% relative to the official answers. After running densification, the system retrieves 78 novel results, containing 23 valid answers, more than half of the missing answers. Not all of the additional 55 results retrieved are necessarily incorrect because GikiCLEF used open-ended questions. Overall, recall is increased to 86.44% (by 19.5 points) and precision falls to 47.22% (by 10.02 points). This suggests that densification does indeed help find additional answers, but the semantic constraints used by the system need to be more precise. For example, for the GC09-09 topic *Name places where Goethe fell in love*, the system does not find additional correct answers, but instead retrieves more incorrect ones because the constraint verifier is too generic: it validates all cities that are mentioned in the same paragraph as the trigger word 'love'.

Densification is more computationally intensive than WikipediaMiner, which achieves the state-of-the-art performance for wikification. The same evaluation was employed to determine how many answers can be found if using WM to recover

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Table 12. *Densification: performance impact on* EQUAL*(micro-averaged)*

System	Corr.	Incorr.	Total	P	R	$F_{0.5}$	F_1	F_2
EQUAL	79	59	138	0.572	0.669	0.589	0.617	0.647
+Densif.	102	114	216	0.472	0.864	0.519	0.610	0.741
+WM1	91	81	172	0.529	0.771	0.564	0.627	0.706

additional links in Wikipedia articles. In this case, EQUAL finds only 34 additional results, containing 12 valid answers. Wikification also helps find additional answers, but to a lesser extent, as it usually creates fewer links per document than densification does (see results in Table 12).

Based on these results, densification does help increase recall, but, in the case of EQUAL, it does so at the cost of precision. To fully benefit from the extra information, the QA system needs to use more advanced constraints and to filter candidate answers using improved information extraction techniques. The set of answers that are not reachable by the current system suggests that, in addition to exploring Wikipedia based on links, the system should also use information extraction methods to search for supporting information in the text of an article. This enables the use of multi-argument constraints that require all arguments to be present in the same context, and not just in the same article/paragraph.

6 Conclusions

This paper proposed a new method for identifying and ranking Wikipedia entities relevant to a document: densification, a type of semantic analysis relevant for building NLP applications. Although it shares similarities with established tasks such as wikification and entity linking, it does not aim for strict disambiguation of named entity mentions. The method instead uses a set of given seed topics to produce a robust ranking of potentially relevant topics. It consists of two steps: pruning, which creates a high-recall set of candidates, and selection, which filters this set and computes relevance scores.

The first step, pruning, is a fuzzy disambiguation approach based on point-wise ranking models trained on data extracted from Wikipedia. It uses a semantic relatedness measure between two articles (Milne and Witten 2008a) to identify and discard candidate topics that have little relevance to the initial set of seeds. In the second step, selection, the relevance of a topic is estimated using both the seeds and the remaining candidate topics that are mentioned nearby. Overlapping mentions are also dealt with. As there is no gold standard data set, an extrinsic validation approach is used. The method is able to produce robust results, even if only a tenth of data is used for training.

A small-scale experiment was carried out to determine whether or not human agreement was high enough to warrant the annotation of a gold standard for densification. The results demonstrated moderate agreement and suggested that acceptable agreement levels could be possible, provided adequate guidelines are given to the raters and that training is first carried out. The experiment showed

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

that assessing relatedness is a rather subjective task; therefore, instead of a costly annotation effort, a more effective way to measure performance is via extrinsic evaluation. The rating methodology developed can be used to estimate the precision of densification models using crowd-sourcing.

Another experiment was carried out to measure the performance of the densification method using Wikipedia. For each article, a random set of links were considered as seeds while the remainder were used to estimate recall. This setting resembles that of wikification, so the performance was compared against several wikification systems. On average, the method proposed achieves the best results. The advantage of this method is that it can be run on a large scale with no annotation required, but its disadvantage is that it cannot be used to estimate precision for densification.

Experiments show that this method can improve the recall of the semantic QA system EQUAL. Densification was used to process Wikipedia articles to identify additional links to relevant topics that are not present in the original markup. The micro-averaged recall was increased from 66.94% to 86.44% (19.5 points).This outperforms the improvement achieved when using new links predicted by the best performing wikification system, WikipediaMiner, which increases micro-averaged recall up to 77.1% (10.2 points).

In future work, the densification system will be applied to other NLP tasks. A natural direction is to investigate whether the system can provide background knowledge for cross-document co-reference resolution. Since it relies on the Wikipedia link graph, the system could also be employed in other languages by exploiting inter-lingual links in Wikipedia.

References

Bentivogli, L., Forner, P., Giuliano, C., Marchetti, A., Pianta, E., and Tymoshenko, K. 2010. Extending English ACE 2005 Corpus annotation with ground-truth links to Wikipedia. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, Beijing, China, August, pp. 19–27. Coling 2010 Organizing Committee.

Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* **3**: 993–1022.

Bryl, V., Giuliano, C., Serafini, L., and Tymoshenko, K. 2010. Supporting natural language processing with background knowledge: coreference resolution case. In P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, and B. Glimm (eds.), *The Semantic Web — ISWC 2010* (9th International Semantic Web Conference, Shanghai, China, Revised Selected Papers, Part I, volume 6496 of Lecture Notes in Computer Science), pp. 80–95. Berlin: Springer.

Bunescu, R. C., and Pasca, M. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, Trento, Italy, April, pp. 9–16. Association for Computational Linguistics.

Chu-Carroll, J., Czuba, K., Prager, J., and Ittycheriah, A. 2003. In question answering, two heads are better than one. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 24–31. Morristown, NJ: Association for Computational Linguistics.

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Cilibrasi, R., and Vitányi, P. M. B. 2007. The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering* **19**(3): 370–83.

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

Cucerzan, S. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, June, pp. 708–16. Association for Computational Linguistics.

Damljanovic, D., and Bontcheva, K. 2012. Named entity disambiguation using linked data. In *9th Extended Semantic Web Conference (ESWC2012)*, Heraklion, Greece.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41**(6): 391–407.

Dornescu, I. 2010. Semantic QA for encyclopaedic questions: EQUAL in GikiCLEF. In C. Peters, G. M. Di Nunzio, M. Kurimo, T. Mandl, and D. Mostefa (eds.), *Multilingual Information Access Evaluation I. Text Retrieval Experiments* (vol. 6241, Lecture Notes in Computer Science), pp. 326–33. Berlin: Springer.

Dornescu, I. 2012. *Encyclopaedic Question Answering*. PhD thesis. Wolverhampton: University of Wolverhampton, UK.

Dredze, M., McNamee, P., Rao, D., Gerber, A., and Finin, T. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pp. 277–285. Stroudsburg, PA: Association for Computational Linguistics.

Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* **76**(5): 378–82.

Gabrilovich, E., and Markovitch, S. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the Twentieth International Joint Conference for Artificial Intelligence*, Hyderabad, India, pp. 1606–11.

Gottipati, S., and Jiang, J. 2011. Linking entities to a knowledge base with query expansion. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, Edinburgh, Scotland, pp. 804–813. Association for Computational Linguistics.

Han, X., and Sun, L. 2011. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – Volume 1*, HLT '11, pp. 945–954. Stroudsburg, PA: Association for Computational Linguistics.

Harabagiu, S., Maiorano, S., and Pasca, M. 2003. Open-domain textual question answering techniques. *Natural Language Engineering* **9**(3): 231–67.

Harabagiu, S., and Moldovan, D. 2003. Question answering. In R. Mitkov (ed.), *The Oxford Handbook of Computational Linguistics*, pp. 560–82. New York: Oxford University Press.

Hatcher, E., and Gospodnetic, O. 2004. *Lucene in Action*. Stanford, CT: Manning.

Hirschman, L., and Gaizauskas, R. 2001. Natural language question answering: the view from here. *Natural Language Engineering* **7**(4): 275–300.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, pp. 50–7. Association for Computing Machinery.

Hovy, E., Gerber, L., Hermjakob, U., Lin, C.-Y., and Ravichandran, D. 2001. Toward semantics-based answer pinpointing. In *HLT '01: Proceedings of the First International Conference on Human Language Technology Research*, pp. 1–7. Morristown, NJ: Association for Computational Linguistics.

Jijkoun, V., Hofmann, K., Ahn, D., Khalid, M. A., van Rantwijk, J., de Rijke, M., and Sang, E. F. T. K.. 2007. The university of Amsterdam's question answering system at QA@CLEF 2007. In C. Peters, V. Jijkoun, T. Mandl, H. Müller, D. W. Oard, A. Peñas, V. Petras, and

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

D. Santos (eds.), *CLEF* (vol. 5152, Lecture Notes in Computer Science), pp. 344–51. Berlin: Springer.

Kanerva, P., Kristofersson, J., and Holst, A. 2000. Random indexing of text samples for latent semantic analysis. In L. Gleitman and A. Josh (eds.), *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, vol. 1036, pp. 1036–7. Pennsylvania, PA: University of Pennsylvania.

Krippendorff, K. 2004. *Content Analysis: An Introduction to Its Methodology*. Thousand Oaks, CA: Sage.

Lang, K. 1995. Newsweeder: learning to filter netnews. In A. Prieditis and S. J. Russell (eds.), *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 331–9. San Francisco, CA: Morgan Kaufmann.

Li, C., Sun, A., and Datta, A. 2013. TSDW: two-stage word sense disambiguation using Wikipedia. *Journal of the American Society for Information Science and Technology* **64**(6): 1203–23.

Li, F., Zheng, Z., Bu, F., Tang, Y., Zhu, X., and Huang, M 2009. THU QUANTA at TAC 2009 KBP and RTE Track. In *Proceedings of the 2009 Text Analysis Conference*, Gaithersburg, MD, November. National Institute of Standards and Technology.

Li, X., and Roth, D. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics-Volume 1*, pp. 1–7. Stroudsburg, PA: Association for Computational Linguistics.

McNamee, P. 2009. Overview of the TAC 2009 knowledge base population track. In *Proceedings of the 2009 Text Analysis Conference*, Gaithersburg, MD, November. National Institute of Standards and Technology.

McNamee, P., Dredze, M., Gerber, A., Garera, N., Finin, T., Mayfield, J., Piatko, C., Rao, D., Yarowsky, D., and Dreyer, M. 2009. HLTCOE approaches to knowledge base population at TAC 2009. In *Proceedings of the 2009 Text Analysis Conference*, Gaithersburg, MD, November. National Institute of Standards and Technology.

McNamee, P., Stoyanov, V., Mayfield, J., Finin, T., Oates, T., Xu, T., Oard, D., and Lawrie, D. 2012. HLTCOE Participation at TAC 2012: entity linking and cold start knowledge base construction. In *Proceedings of the Fifth Text Analysis Conference (TAC 2012)*, Gaithersburg, MD, November. National Institute of Standards and Technology.

Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. 2011. DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, pp. 1–8. New York: Association for Computing Machinery.

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 2007)*, pp. 233–242. New York: Association for Computing Machinery.

Milne, D., and Witten, I. H. 2008a. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceedings of the first AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008)*, pp. 25–30. Menlo Park, CA: AAAI Press.

Milne, D., and Witten, I. H. 2008b. Learning to link with Wikipedia. In J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K.-S. Choi, and A. Chowdhury (eds.), *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008)*, pp. 509–518. New York: Association for Computing Machinery.

Moldovan, D., Clark, C., Harabagiu, S., and Hodges, D. 2007. Cogex: a semantically and contextually enriched logic prover for question answering. *Journal of Applied Logic* **5**(1): 49–69.

Rizzo, G., and Troncy, R. 2011. NERD: evaluating named entity recognition tools in the web of data. In *Proceedings of the ISWC'11 Workshop on Web Scale Knowledge Extraction (WEKEX'11)*, Bonn, Germany, pp. 1–16.

Sahlgren, M. 2005. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE*, Copenhagen, Denmark.

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press

Santos, D., and Cabral, L. M. 2009. GikiCLEF: expectations and lessons learned. In C. Peters, G. M. Di Nunzio, M. Kurimo, D. Mostefa, A. Peñas, and G. Roda (eds.), *CLEF 1* (vol. 6241, Lecture Notes in Computer Science), pp. 212–22. Berlin: Springer.

Santos, D., Cardoso, N., Carvalho, P., Dornescu, I., Hartrumpf, S., Leveling, J., and Skalban, Y. 2009. GikiP at GeoCLEF 2008: joining GIR and QA forces for querying Wikipedia. In C. Peters, T. Deselaers, N. Ferro, J. Gonzalo, A. Peñas, G. J. F. Jones, M. Kurimo, T. Mandl, and V. Petras (eds.), *Proceedings of the 9th Cross-Language Evaluation Forum Conference on Evaluating Systems for Multilingual and Multimodal Information Access* (vol. 5706, Lecture Notes in Computer Science), pp. 894–905. Berlin: Springer.

Schlaefer, N., Ko, J., Betteridge, J., Sautter, G., Pathak, M., and Nyberg, E. 2007. Semantic extensions of the Ephyra QA system for TREC 2007. In E. M. Voorhees and L. P. Buckland (eds.), *Proceedings of the Sixteenth Text REtrieval Conference (TREC)*, Gaithersburg, MD: National Institute of Standards and Technology.

Scott, W. A. 1955. Reliability of content analysis: the case of nominal scale coding. *Public Opinion Quarterly* **19**(3): 321–5.

Sim, J., and Wright, C. C. 2005. The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical Therapy* **85**(3): 257–68.

Spearman, C. 1904. The proof and measurement of association between two things. *American Journal of Psychology* **15**(1): 72–101.

Voorhees, E. M. 2001. The TREC question answering track. *Natural Language Engineering* **7**(4): 361–78.

https://doi.org/10.1017/S1351324913000296 Published online by Cambridge University Press