DBpedia Spotlight: Shedding Light on the Web of Documents

Pablo N. Mendes[1], Max Jakob[1], Andrés García-Silva[2], Christian Bizer[1]

[1]Web-based Systems Group, Freie Universität Berlin, Germany
first.last@fu-berlin.de
[2]Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
hgarcia@fi.upm.es

ABSTRACT

Interlinking text documents with Linked Open Data enables the Web of Data to be used as background knowledge within document-oriented applications such as search and faceted browsing. As a step towards interconnecting the Web of Documents with the Web of Data, we developed DBpedia Spotlight, a system for automatically annotating text documents with DBpedia URIs. DBpedia Spotlight allows users to configure the annotations to their specific needs through the DBpedia Ontology and quality measures such as prominence, topical pertinence, contextual ambiguity and disambiguation confidence. We compare our approach with the state of the art in disambiguation, and evaluate our results in light of three baselines and six publicly available annotation systems, demonstrating the competitiveness of our system. DBpedia Spotlight is shared as open source and deployed as a Web Service freely available for public use.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language parsing and understanding*; I.7 [**Document and Text Processing**]: [Miscellaneous]

General Terms

Algorithms, Experimentation

Keywords

Text Annotation, Linked Data, DBpedia, Named Entity Disambiguation

1. INTRODUCTION

As the Linked Data ecosystem develops [3], so do the mutual rewards for structured and unstructured data providers alike. Higher interconnectivity between information sources has the potential of increasing discoverability, reusability, and hence the utility of information. By connecting unstructured information in text documents with Linked Data, facts from the Web of Data can be used, for instance, as complementary information on web pages to enhance information retrieval, to enable faceted document browsing [15] and customization of web feeds based on semantics [18].

DBpedia [4] is developing into an interlinking hub in the Web of Data that enables access to many data sources in the Linked Open Data cloud. It contains encyclopedic knowledge from Wikipedia for about 3.5 million resources. About half of the knowledge base is classified in a consistent cross-domain ontology with classes such as persons, organisations or populated places; as well as more fine-grained classifications like basketball player or flowering plant. Furthermore, it provides a rich pool of resource attributes and relations between the resources, connecting products to their makers, or CEOs to their companies, for example.

In order to enable the linkage of Web documents with this hub, we developed DBpedia Spotlight, a system to perform annotation of DBpedia resources mentioned in text. In the annotation task, the user provides text fragments (documents, paragraphs, sentences) and wishes to identify URIs for resources mentioned within that text. One of the main challenges in annotation is ambiguity: an entity name, or surface form, may be used in different contexts to refer to different DBpedia resources. For example, the surface form '*Washington*' can be used to refer the resources `dbpedia:George_Washington`, `dbpedia:Washington,_D.C.` and `dbpedia:Washington_(U.S._state)` (among others). For human readers, the disambiguation, i.e. the decision between candidates for an ambiguous surface form, is usually performed based on the readers' knowledge and the context of a concrete mention. However, the automatic disambiguation of entity mentions remains a challenging problem.

The goal of DBpedia Spotlight is to provide an adaptable system to find and disambiguate natural language mentions of DBpedia resources. Much research has been devoted to the problem of automatic disambiguation - as we discuss in Section 5. In comparison with previous work, DBpedia Spotlight aims at a more comprehensive and flexible solution. First, while other annotation systems are often restricted to a small number of resource types, such as people, organisations and places, our system attempts to annotate DBpedia resources of any of the 272 classes (more than 30 top level) in the DBpedia Ontology. Second, since a single generic solution is unlikely to fit all task-specific require-

I-SEMANTICS 2011, 7th Int. Conf. on Semantic Systems, Sept. 7-9, 2011, Graz, Austria

ments, our system enables user-provided configurations for different use cases with different needs. Users can flexibly specify the domain of interest, as well as the desired coverage and error tolerance for each of their specific annotation tasks.

DBpedia Spotlight can take full advantage of the DBpedia ontology for specifying which concepts should be annotated. Annotations can be restricted to instances of specific classes (or sets of classes) including subclasses. Alternatively, arbitrary SPARQL queries over the DBpedia knowledge base can be provided in order to determine the set of instances that should be annotated. For instance, consider use cases where users have prior knowledge of some aspects of the text (e.g. dates), and have specific needs for the annotations (e.g. only Politicians). A SPARQL query can be sent to DBpedia Spotlight in order to constrain the annotated resources to only politicians in office between 1995 and 2000, for instance. In general, users can create restrictions using any part of the DBpedia knowledge base.

Moreover, DBpedia Spotlight computes scores such as prominence (how many times a resource is mentioned in Wikipedia), topical relevance (how close a paragraph is to a DBpedia resource's context) and contextual ambiguity (is there more than one candidate resource with similarly high topical relevance for this surface form in its current context?). Users can configure these parameters according to their task-specific requirements.

We evaluate DBpedia Spotlight in two experiments. First we test our disambiguation strategy on thousands of unseen (held out) DBpedia resource mentions from Wikipedia. Second, we use a set of manually annotated news articles in order to compare our annotation with publicly available annotation services.

DBpedia Spotlight is deployed as a Web Service, and features a user interface for demonstration. The source code is publicly available under the Apache license V2, and the documentation is available at `http://dbpedia.org/spotlight`.

In Section 2 we describe our approach, followed by an explanation of how our system can be used (Section 3). In Section 4 we present our evaluation methodology and results. In Section 5 we discuss related work and in Section 6 we present our conclusions and future work.

2. APPROACH

Our approach works in four-stages. The *spotting* stage recognizes in a sentence the phrases that may indicate a mention of a DBpedia resource. *Candidate selection* is subsequently employed to map the spotted phrase to resources that are candidate disambiguations for that phrase. The *disambiguation* stage, in turn, uses the context around the spotted phrase to decide for the best choice amongst the candidates. The annotation can be customized by users to their specific needs through *configuration* parameters explained in subsection 2.5. In the remainder of this section we describe the datasets and techniques used to enable our annotation process.

2.1 Dataset and Notation

We utilize the graph of labels, redirects and disambiguations in DBpedia to extract a lexicon that associates multiple surface forms to a resource and interconnects multiple resources to an ambiguous label. *Labels* of the DBpedia resources are created from Wikipedia page titles, which can be seen as community-approved surface forms. *Redirects* to URIs indicate synonyms or alternative surface forms, including common misspellings and acronyms. Their labels also become surface forms. *Disambiguations* provide ambiguous surface forms that are "confusable" with all resources they link to. Their labels become surface forms for all target resources in the disambiguation page. Note that we erase trailing parentheses from the labels when constructing surface forms. For example the label '*Copyright (band)*' produces the surface form '*Copyright*'. This means that labels of resources and of redirects can also introduce ambiguous surface forms, additionally to the labels coming from titles of disambiguation pages. The collection of surface forms created as a result constitutes a controlled set of commonly used labels for the target resources.

Another source of textual references to DBpedia resources are *wikilinks*, i.e. the page links in Wikipedia that interconnect the articles. We pre-processed Wikipedia articles, extracting every wikilink $l = (s, r)$ with surface form s as anchor text and resource r as link target, along with the paragraph representing the context of that wikilink occurrence. Each wikilink was stored as an evidence of occurrence $o = (r, s, C)$. Each occurrence o records the fact that the DBpedia resource r represented by the link target has been mentioned in the context of the paragraph through the use of the surface form s. Before storage, the context paragraph was tokenized, stopworded and stemmed, generating a vector of terms $W = \langle w_1, ..., w_n \rangle$. The collection of occurrences for each resource was then stored as a document in a Lucene index[1] for retrieval in the disambiguation stage.

Wikilinks can also be used to estimate the likelihood of a surface form s referring to a specific candidate resource $r \in R_s$. We consider each wikilink as evidence that the anchor text is a commonly used surface form for the DBpedia resource represented by the link target. By counting the number of times a surface form occurred with and without a DBpedia resource $n(s, r)$, we can empirically estimate a prior probability of seeing a resource r given that surface form s was used $P(r|s) = n(s, r)/n(s)$.

2.2 Spotting Algorithm

We use the extended set of labels in the lexicalization dataset to create a lexicon for spotting. The implementation used was the LingPipe Exact Dictionary-Based Chunker [2] which relies on the Aho-Corasick string matching algorithm [1] with longest case-insensitive match.

Since for many use cases it is unnecessary to annotate common words, a configuration flag can instruct the system to disregard in this stage any spots that are only composed of verbs, adjectives, adverbs and prepositions. The part of speech tagger used was the LingPipe implementation based on Hidden Markov Models.

2.3 Candidate Selection

We follow the spotting with a candidate selection stage in order to map resource names to candidate disambiguations (e.g. *Washington* as reference to a city, to a person or to a state). We use the DBpedia Lexicalization dataset for determining candidate disambiguations for each surface form.

The candidate selection offers a chance to narrow down the space of disambiguation possibilities. Selecting fewer

[1] `http://lucene.apache.org`

candidates can increase time performance, but it may reduce recall if performed too aggressively. Due to our generality and flexibility requirements, we decided to employ minimal pre-filtering and postpone the selection to a user-configured post-disambiguation configuration stage. Other approaches for candidate selection are within our plans for future work.

The candidate selection phase can also be viewed as a way to pre-rank the candidates for disambiguation before observing a surface form in the context of a paragraph. Choosing the DBpedia resource with highest prior probability for a surface form is the equivalent of selecting the "default sense" of some phrase according to its usage in Wikipedia. The prior probability scores of the lexicalizations dataset, for example, can be utilized at this point. We report the results for this approach as a baseline in Section 4.

2.4 Disambiguation

After selecting candidate resources for each surface form, our system uses the context around the surface forms, e.g. paragraphs, as information to find the most likely disambiguations.

We modeled DBpedia resource occurrences in a Vector Space Model (VSM) [22] where each DBpedia resource is a point in a multidimensional space of words. In light of the most common use of VSMs in Information Retrieval (IR), our representation of a DBpedia resource is the analogous of a document containing the aggregation of all paragraphs mentioning that concept in Wikipedia. Similarly, the TF (Term Frequency) weight is commonly used in IR to measure the local relevance of a term in a document. In our model, TF represents the relevance of a word for a given resource. In addition, the Inverse Document Frequency (IDF) weight [16] represents the general importance of the word in the collection of DBpedia resources.

Albeit successful for document retrieval, the IDF weight fails to adequately capture the importance of a word for disambiguation. For the sake of illustration, suppose that the term '*U.S.A*' occurs in only 3 concepts in a collection of 1 million concepts. Its IDF will be very high, as its document frequency is very low (3/1,000,000). Now suppose that the three concepts with which it occurs are `dbpedia:Washington,_D.C.`, `dbpedia:George_Washington`, and `dbpedia:Washington_(U.S._State)`. As it turns out, despite the high IDF weight, the word '*U.S.A*' would be of little value to disambiguate the surface form '*Washington*', as all three potential disambiguations would be associated with that word. IDF gives an insight into the global importance of a word (given all resources), but fails to capture the importance of a word for a specific set of candidate resources.

In order to weigh words based on their ability to distinguish between candidates for a given surface form, we introduce the Inverse Candidate Frequency (ICF) weight. The intuition behind ICF is that the discriminative power of a word is inversely proportional to the number of DBpedia resources it is associated with. Let R_s be the set of candidate resources for a surface form s. Let $n(w_j)$ be the total number of resources in R_s that are associated with the word w_j. Then we define:

$$ICF(w_j) = \log \frac{|R_s|}{n(w_j)} = \log |R_s| - \log n(w_j) \qquad (1)$$

The theoretical explanation for ICF is analogous to Deng et al. [9], based on Information Theory. Entropy [23] has been commonly used to measure uncertainty in probability distributions. It is argued that the discriminative ability of a context word should be inversely proportional to the entropy, i.e. a word commonly co-occurring with many resources is less discriminative overall. With regard to a word's association with DBpedia resources, the entropy of a word can be defined as: $E(w) = -\sum_{i \in R_s} P(r_i|w) \log P(r_i|w)$. Suppose that the word w is connected to those resources with equal probability $P(r|w) = 1/n(w)$, the maximum entropy is transformed to $E(w) = \log n(w)$. Since generally the entropy tends to be proportional to the frequency $n(w)$, we use the maximum entropy to approximate the exact entropy in the ICF formula. This simplification has worked well in our case, simplifying the calculations and reducing storage and search time requirements.

Given the VSM representation of DBpedia resources with TF*ICF weights, the disambiguation task can be cast as a ranking problem where the objective is to rank the correct DBpedia resource at position 1. Our approach is to rank candidate resources according to the similarity score between their context vectors and the context surrounding the surface form. In this work we use cosine as the similarity measure.

2.5 Configuration

Many of the current approaches for annotation tune their parameters to a specific task, leaving little flexibility for users to adapt their solution to other use cases. Our approach is to generate a number of metrics to inform the users and let them decide on the policy that best fits their needs. In order to decide whether to annotate a given resource, there are several aspects to consider: can this resource be confused easily with another one in the given context? Is this a commonly mentioned resource in general? Was the disambiguation decision made with high confidence? Is the resource of the desired type? Is the resource in a complex relationship within the knowledge base that rules it out for annotation? The offered configuration parameters are described next.

Resource Set to Annotate. The use of DBpedia resources as targets for annotation enables interesting flexibility. The simplest and probably most widely used case is to annotate only resources of a certain type or set of types. In our case the available types are derived from the class hierarchy provided by the DBpedia Ontology. Users can provide whitelists (allowed) or blacklists (forbidden) of URIs for annotation. Whitelisting a class will allow the annotation of all direct instances of that class, as well as all instances of subclasses. Support for SPARQL queries allows even more flexibility by enabling the specification of arbitrary graph patterns. There is no restriction to the complexity of relationships that a resource must fulfil in this configuration step. For instance, the user could choose to only annotate concepts that are related to a specific geographic area, time period in history, or are closely connected within the Wikipedia category system.

Resource Prominence. For many applications, the annotation of rare or exotic resources is not desirable. For example, the `Saxon_genitive` (*'s*) is very commonly found in English texts to indicate possession (e.g. Austria*'s* mountains are beautiful), but it can be argued that for many use cases its

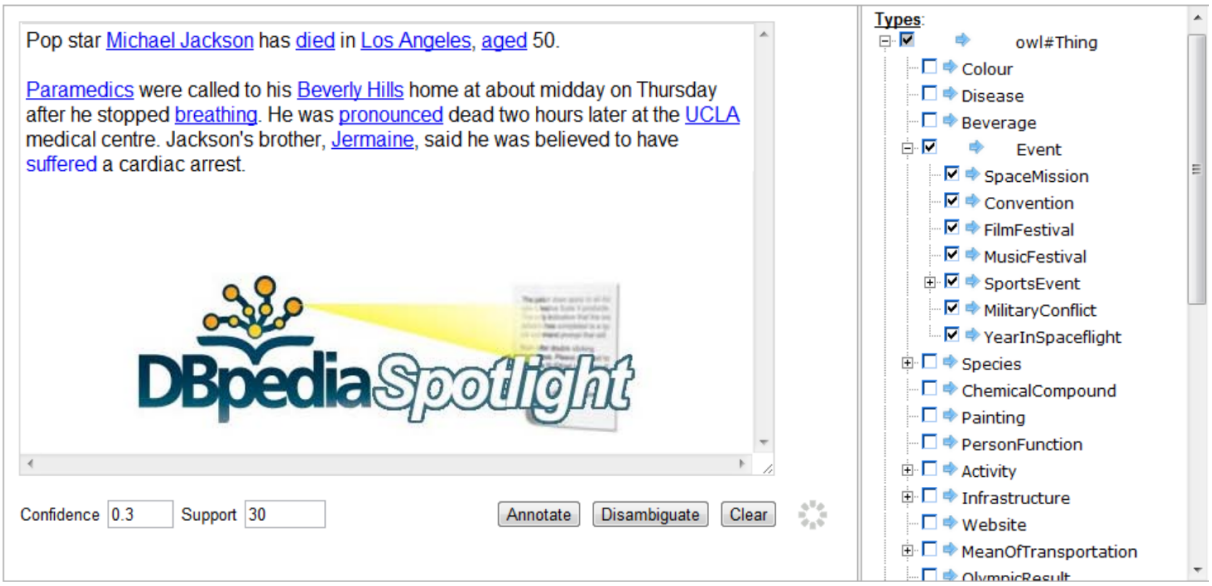

Figure 1: DBpedia Spotlight Web Application.

annotation is rather uninformative. An indicator for that is that it has only seven Wikipedia inlinks. With the *support parameter*, users can specify the minimum number of inlinks a DBpedia resource has to have in order to be annotated.

Topic Pertinence. The topical relevance of the annotated resource for the given context can be measured by the similarity score returned by the disambiguation step. The score is higher for paragraphs that match more closely the recorded observations for a DBpedia resource. In order to constrain annotations to topically related resources, a higher threshold for the topic pertinence can be set.

Contextual Ambiguity. If more than one candidate resource has high topical pertinence to a paragraph, it may be harder to disambiguate between those resources because they remain partly ambiguous in that context. The difference in the topical relevance of two candidate resources to a paragraph gives us an insight on how "confused" the disambiguation step was in choosing between these resources. The score is computed by the relative difference in topic score between the first and the second ranked resource. Applications that require high precision may decide to reduce risks by not annotating resources when the contextual ambiguity is high.

Disambiguation Confidence. We define a confidence parameter, ranging from 0 to 1, of the annotation performed by DBpedia Spotlight. This parameter takes into account factors such as the topical pertinence and the contextual ambiguity. Setting a high confidence threshold instructs DBpedia Spotlight to avoid incorrect annotations as much as possible at the risk of losing some correct ones. We estimated this parameter on a development set of 100,000 Wikipedia samples. The rationale is that a confidence value of 0.7 will eliminate 70% of incorrectly disambiguated test cases. For example, given a confidence of 0.7, we get the topical pertinence threshold that 70% of the wrong test samples are below. We integrate that with the contextual ambiguity score by requiring a low ambiguity when the confidence is high. A confidence of 0.7, therefore, will only annotate resources if the contextual ambiguity is less than $(1-confidence) = 0.3$. We address the adequacy of this parameter in our evaluation.

3. USING DBPEDIA SPOTLIGHT

DBpedia Spotlight is available both as a Web Service and via a Web Application. In addition, we have published the lexicalization dataset in RDF so that the community can benefit from the collected surface forms and the DBpedia resources representing their possible meanings.

3.1 Web Application

By using the Web application, users can test and visualize the results of the different service functions. The interface allows users to configure confidence, support, and to select the classes of interest from the DBpedia ontology. Text can be entered in a text box and, at user's request, DBpedia Spotlight will highlight the surface forms and create associations with their corresponding DBpedia resources. Figure 1 shows an example of a news article snippet after being annotated by our system. In addition to *Annotate*, we offer a *Disambiguate* operation where users can request the disambiguation of selected phrases (enclosed in double square brackets). In this case, our system bypasses the spotting stage and annotates only the selected phrases with DBpedia resources. This function is useful for user interfaces that allow users to mouse-select text, as well as for the easy incorporation of our disambiguation step into third-party applications that already perform spotting.

3.2 Web Service

In order to facilitate the integration of DBpedia Spotlight into external web processes, we implemented RESTful and SOAP web services for the annotation and disambiguation processes. The web service interface allows access to both the Annotate and the Disambiguate operations and to all the configuration parameters of our approach. Thus, in addition to confidence, support and DBpedia classes, we accept SPARQL queries for the DBpedia knowledge base to select

the set of resources that are going to be used when annotating. These web services return HTML, XML, JSON or XHTML+RDFa documents where each DBpedia resource identified in the text is related to the text chunk where it was found. The XML fragment presented below shows part of the annotation of the news snippet shown in Figure 1.

```
<Annotation text="Pop star Michael Jackson..."
   confidence="0.3" support="30"
   types="Person,Place,...">
   <Resources>
     <Resource URI="dbpedia:Michael_Jackson"
       support="5761"
       types="MusicalArtist,Artist,Person"
       surfaceForm="Michael Jackson" offset="9"
       similarityScore="0.31504717469215393" />
     ...
   </Resources>
</Annotation>
```

Figure 2: Example XML fragment resulting from the annotation service.

3.3 Lexicalization dataset

Besides the DBpedia Spotlight system, the data produced in this work is also shared in a format to ease its consumption in a number of use cases. The dataset described in Section 2.1 was encoded in RDF using the Lexvo vocabulary [8] and is provided for download as a DBpedia dataset. We use the property `lexvo:label` rather than `rdfs:label` or `skos:altLabel` to associate a resource with surface form strings. The `rdfs:label` property intends to represent "a human-readable version of a resource's name"[2]. The SKOS Vocabulary "enables a distinction to be made between the preferred, alternative and 'hidden' lexical labels" through their `skos:prefLabel` and `skos:altLabel`. The DBpedia Spotlight dataset does not claim that a surface form is the name of a resource, and neither intends to assert preference between labels. Hence, we use `lexvo:label` in order to describe the resource - surface form association with regard to actual language use. Association scores (e.g. prior probabilities) are attached to `lexvo:label` relationships through named graphs.

Users interested in finding names, alternative or preferred labels can use the provided information in order to make an informed task-specific choice. Imagine a user attempting to find DBpedia URIs for presidents and colleges in his company's legacy database. The table called President contains two columns: last name, alma mater. Users may use a SPARQL query, for example, to select the default sense for the surface form '*Bush*', given that it is known it has a relationship with the surface form '*Harvard Business*' [3]. The lexicalizations dataset will provide links between alternative spellings (e.g. '*Bush*' → `dbpedia:George_W._Bush`) and the knowledge base (DBpedia) will provide the background knowledge connecting the resource `dbpedia:George_W._Bush` to his alma mater `dbpedia:Harvard_Business_School`. The association scores will help to rank the most likely of the candidates in this context.

The dataset can also be used to get information about the strength of association between a surface form and a resource, term ambiguity or the default sense of a surface form, just to cite a few use cases.

4. EVALUATION

We carried out two evaluations of DBpedia Spotlight. A large scale automatic evaluation tested the performance of the disambiguation component in choosing the correct candidate resources for a given surface form. In order to provide an evaluation of the whole system in a specific annotation scenario, we also carried out an experiment using a manually annotated test corpus. In that evaluation we compare our results with those of several publicly available annotation services.

4.1 Disambiguation Evaluation

Wikipedia provides a wealth of annotated data that can be used to evaluate our system on a large scale. We randomly selected 155,000 wikilink samples and set aside as test data. In order to really capture the ability of our system to distinguish between multiple senses of a surface form, we made sure that all these instances have ambiguous surface forms. We used the remainder of the samples collected from Wikipedia (about 69 million) as DBpedia resource occurrences providing context for disambiguation as described in Section 2.

In this evaluation, we were interested in the performance of the disambiguation stage. A spotted surface form, taken from the anchor text of a wikilink, is given to the disambiguation function[4] along with the paragraph that it was mentioned in. The task of the disambiguation service is to select candidate resources for this surface form and decide between them based on the context.

In order to better assess the contribution of our approach, we included three baseline methods:

- *Random Baseline* performs candidate selection and picks one of the candidates with uniform probability. This baseline serves as a control for easy disambiguations, since for low ambiguity terms, even random choice should perform reasonably.
- *Default Sense Baseline* performs candidate selection and chooses the candidate with the highest prior probability (without using the context). More formally: $\arg\max_{r \in R_s} P(r|s)$. This baseline helps to assess how common were the DBpedia resources included in the annotation dataset.
- *Default Similarity* uses TF*IDF term scoring as a reference to evaluate the influence of our TF*ICF approach.

4.1.1 Results

The results for the baselines and DBpedia Spotlight are presented in Table 1. The performance of the baseline that makes random disambiguation choices confirms the high ambiguity in our dataset (less than 1/4 of the disambiguations were correct at random). Using the prior probability to choose the default sense performs reasonably well, being accurate in 55.12% of the disambiguations. This is indication

[2] `http://www.w3.org/TR/rdf-schema/`

[3] The SPARQL formulation for this query and other examples are available from the project page.

[4] in our implementation, for convenience, the candidate selection can be called from the disambiguation

Disambiguation Approach	*Accuracy*
Baseline Random	17.77%
Baseline Default Sense	55.12%
Baseline TF*IDF	55.91%
DBpedia Spotlight TF*ICF	73.39%
DBpedia Spotlight Mixed	80.52%

Table 1: Accuracies for each of the approaches tested in the disambiguation evaluation.

that our evaluation set was composed by a good balance of common DBpedia resources and less prominent ones. The use of context for disambiguation through the default scoring of TF*IDF obtained 55.91%, while the TF*ICF score introduced in this work improved the results to 73.39%.

The performance of TF*ICF is an encouraging indication that a simple ranking-based disambiguation algorithm can be successful if enough contextual evidence is provided.

We also attempted a simple combination of the prior (default sense) and TF*ICF scores, which we called DBpedia Spotlight Mixed. The mixing weights were estimated through a small experiment using linear regression over held out training data. The results reported in this work used mixed scores computed through the formula:

$$\begin{aligned} Mixed(r, s, C) = \\ & 1234.3989 * P(r|s) \\ & +0.9968 * contextualScore(r, s, C) \\ & -0.0275 \end{aligned} \quad (2)$$

The prior probability $P(r|s)$ was calculated as described in Section 2.1. The contextual score used was the cosine similarity of term vectors weighted by TF*ICF as described in Section 2.4. Further research is needed to carefully examine the contribution of each component to the final score.

4.2 Annotation Evaluation

Although we used an unseen dataset for evaluating our disambiguation approach, it is possible that the type of discourse and the annotation style of Wikipedia would bias the results in favor of systems trained with that kind of data. The Wikipedia guidelines for link creation focus on non-obvious references[5]. If a wikilink would not contribute to the understanding of a specific article, the Wikipedia Manual of Style discourages its creation. Therefore, we created a manual evaluation dataset from a news corpus in order to complement that evaluation. In this second evaluation, we would like to assess completeness of linking as well. We created an annotation scenario in which the annotators were asked to add links to DBpedia resources for all phrases that would add information to the provided text.

Our test corpus consisted of 35 paragraphs from New York Times documents from 8 different categories. In order to construct a gold standard, each evaluator first independently annotated the corpus, after which they met and agreed upon the ground truth evaluation choices. The ratio of annotated to not-annotated tokens was 33%. This corpus is described in more details on the project homepage.

[5] `http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_(linking)`

We compared our results on this test corpus with the performance of publicly available annotation services: OpenCalais[6], Zemanta[7], Ontos Semantic API[8], The Wiki Machine[9], Alchemy API[10] and M&W's wikifier [20]. Linking to DBpedia is supported in those services in different levels. Alchemy API provides links to DBpedia and Freebase among other sources. Open Calais and Ontos provide some limited linkage between their private identifiers and DBpedia resources. As of the time of writing, Ontos only links people and companies to DBpedia. For the cases where the systems were able to extract resources but do not give DBpedia URIs, we used a simple transformation on the extracted resources that constructed DBpedia URIs from labels - e.g. 'apple' becomes `dbpedia:Apple`. We report results with and without this transformation. The results that used the transformation are labeled Ontos+Naïve and Open Calais+Naïve. The service APIs of Zemanta, The Wiki Machine and M&W do not explicitly return DBpedia URIs, but the URIs can be inferred from the Wikipedia links that they return.

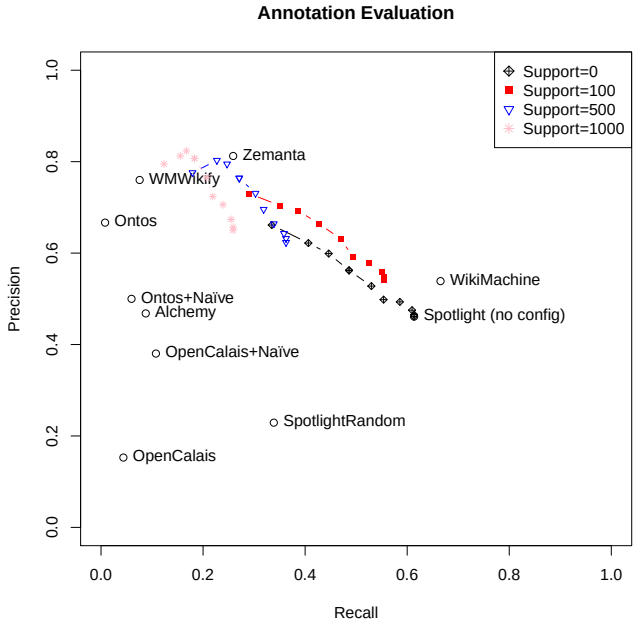

Figure 3: DBpedia Spotlight with different configurations (lines) in comparison with other systems (points).

4.2.1 Results

Retrieval as well as classification tasks exhibit an inherent precision-recall trade-off [5]. The configuration of DBpedia Spotlight allows users to customize the level of annotation to their specific application needs. Figure 3 shows the evaluation results. Each point in the plot represents the precision (vertical axis) and recall (horizontal axis) of each evaluation

[6] `http://www.opencalais.com`
[7] `http://www.zemanta.com`
[8] `http://www.ontos.com`
[9] `http://thewikimachine.fbk.eu`
[10] `http://www.alchemyapi.com`

System	*F1*
DBpedia Spotlight (best configuration)	56.0%
DBpedia Spotlight (no configuration)	45.2%
The Wiki Machine	59.5%
Zemanta	39.1%
Open Calais+Naïve	16.7%
Alchemy	14.7%
Ontos+Naïve	10.6%
Open Calais	6.7%
Ontos	1.5%

Table 2: F_1 scores for each of the approaches tested in the annotation evaluation.

run. The lines show the trade-off between precision and recall as we vary the confidence and support parameters in our service. Each line represents one value of support (varying from 0 to 1000). Each point in the line is a value of confidence (0.1 to 0.9) for the corresponding support. It can be observed that higher confidence values (with higher support) produce higher precision at the cost of some recall and vice versa. This is encouraging indication that our parameters achieve their objectives.

The shape of the displayed graph shows that the performance of DBpedia Spotlight is in a competitive range. Most annotation services lay beneath the F_1-score of our system with every confidence value. Table 5 shows the best F_1-scores of each approach. The best F_1-score of DBpedia Spotlight was reached with confidence value of 0.6. The WikiMachine has the highest F_1-score, but tends to over-annotate the articles, which results in a high recall, at the cost of low precision. Meanwhile, Zemanta dominates in precision, but has low recall. With different confidence and support parameters, DBpedia Spotlight is able to approximate the results of both WikiMachine and Zemanta, while offering many other configurations with different precision-recall trade-offs in between.

5. RELATED WORK

Many existing approaches for entity annotation have focused on annotating salient entity references, commonly only entities of specific types (Person, Organization, Location) [14, 21, 24, 12] or entities that are in the subject of sentences [11]. Hassell et al. [14] exploit the structure of a call for papers corpus for relation extraction and later disambiguation of academic researchers. Rowe [21] concentrates on disambiguating person names with social graphs, while Volz et al. [24] present a disambiguation algorithm for the geographic domain that is based on popularity scores and textual patterns. Gruhl et al. [12] also constrain their annotation efforts to cultural entities in a specific domain. Our objective is to be able to annotate any entities in DBpedia.

Other approaches have attempted the non-type-specific annotation of entities. However, several optimize their approaches for precision, leaving little flexibility for users with use cases where recall is important, or they have not evaluated the applicability of their approaches with more general use cases [10, 6, 7, 19].

SemTag [10] was the first Web-scale named entity disambiguation system. They used metadata associated with each entity in an entity catalog derived from TAP [13] as context for disambiguation. SemTag specialized in precision at the cost of recall, producing an average of less than two annotations per page.

Bunesco and Pasca [6], Cucerzan [7], Mihalcea and Csomai (Wikify!) [19] and Witten and Milne (M&W) [20], like us, also used text from Wikipedia in order to learn how to annotate. Bunesco and Pasca only evaluate articles under the "people by occupation" category, while Cucerzan's and Wikify!'s conservative spotting only annotate 4.5% and 6% of all tokens in the input text, respectively. In Wikify!, this spotting yields surface forms with low ambiguity for which even a random disambiguator achieves an F_1 score of 0.6.

Fader et al. [11] chooses the candidate with the highest prior probability unless the contextual evidence is higher than a threshold. In their dataset 27.94% of the surface forms are unambiguous and 46.53% of the ambiguous ones can be correctly disambiguated by just choosing the default sense (according to our index).

Kulkarni et al. [17] attempts the joint optimization of all spotted surface forms in order to realize the collective annotation of entities. The inference problem formulated by the authors is NP-hard, leading to their proposition of a Linear Programing and a Hill-climbing approach for optimization. We propose instead a simple, inexpensive approach that can be easily configured and adapted to task-specific needs, facilitated by the DBpedia Ontology and configuration parameters.

6. CONCLUSION

In this paper we presented DBpedia Spotlight, a tool to detect mentions of DBpedia resources in text. It enables users to link text documents to the Linked Open Data cloud through the DBpedia interlinking hub. The annotations provided by DBpedia Spotlight enable the enrichment of websites with background knowledge, faceted browsing in text documents and enhanced search capabilities. The main advantage of our system is its comprehensiveness and flexibility, allowing one to configure it based on the DBpedia ontology, as well as prominence, contextual ambiguity, topical pertinence and confidence scores. The resources that should be annotated can be specified by a list of resource types or by more complex relationships within the knowledge base.

We compared our system with other publicly available services and showed how we retained competitiveness with a more configurable approach. In the future we plan to incorporate more knowledge from the Linked Open Data cloud in order to enhance the annotation algorithm.

A project page with news, documentation, downloads, demonstrations and other information is available at `http://dbpedia.org/spotlight`.

7. ACKNOWLEDGEMENTS

The development of DBpedia Spotlight was supported by the European Commission through the project *LOD2 – Creating Knowledge out of Linked Data* and by *Neofonie GmbH*, a Berlin-based company offering technologies in the area of Web search, social media and mobile applications.

Thanks to Andreas Schultz and Paul Kreis for their help with setting up our servers and evaluation, and to Joachim Daiber for his contributions in dataset creation, evaluation clients and preprocessing code that was partially utilized in the finalizing stages of this work.

8. REFERENCES

[1] A. V. Aho and M. J. Corasick. Efficient string matching: an aid to bibliographic search. *Commun. ACM*, 18:333–340, June 1975.

[2] Alias-i. LingPipe 4.0.0. `http://alias-i.com/lingpipe`, retrieved on 24.08.2010, 2008.

[3] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[4] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7:154–165, September 2009.

[5] M. Buckland and F. Gey. The relationship between Recall and Precision. *J. Am. Soc. Inf. Sci.*, 45(1):12–19, January 1994.

[6] R. C. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, 2006.

[7] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, pages 708–716, 2007.

[8] G. de Melo and G. Weikum. Language as a foundation of the Semantic Web. In C. Bizer and A. Joshi, editors, *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC 2008)*, volume 401 of *CEUR WS*, Karlsruhe, Germany, 2008. CEUR.

[9] H. Deng, I. King, and M. R. Lyu. Entropy-biased models for query representation on the click graph. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346, New York, NY, USA, 2009. ACM.

[10] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 178–186, New York, NY, USA, 2003. ACM.

[11] A. Fader, S. Soderland, and O. Etzioni. Scaling wikipedia-based named entity disambiguation to arbitrary web text. In *Proceedings of the WikiAI 09 - IJCAI Workshop: User Contributed Knowledge and Artificial Intelligence: An Evolving Synergy*, Pasadena, CA, USA, July 2009.

[12] D. Gruhl, M. Nagarajan, J. Pieper, C. Robson, and A. P. Sheth. Context and domain knowledge enhanced entity spotting in informal text. In *International Semantic Web Conference*, pages 260–276, 2009.

[13] R. V. Guha and R. McCool. Tap: A semantic web test-bed. *J. Web Sem.*, 1(1):81–87, 2003.

[14] J. Hassell, B. Aleman-Meza, and I. Arpinar. Ontology-driven automatic entity disambiguation in unstructured text. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 44–57. Springer Berlin / Heidelberg, 2006.

[15] M. Hearst. UIs for Faceted Navigation: Recent Advances and Remaining Open Problems. In *Workshop on Computer Interaction and Information Retrieval, HCIR*, Redmond, WA, Oct. 2008.

[16] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[17] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 457–466, New York, NY, USA, 2009. ACM.

[18] P. N. Mendes, A. Passant, P. Kapanipathi, and A. P. Sheth. Linked open social signals. In *Web Intelligence and Intelligent Agent Technology, 2010. WI-IAT '10. IEEE/WIC/ACM International Conference on*, 2010.

[19] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, New York, NY, USA, 2007. ACM.

[20] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 509–518, New York, NY, USA, 2008. ACM.

[21] M. Rowe. Applying semantic social graphs to disambiguate identity references. In L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvï£¡nen, R. Mizoguchi, E. Oren, M. Sabou, and E. Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 461–475. Springer Berlin / Heidelberg, 2009.

[22] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18:613–620, November 1975.

[23] C. E. Shannon. Prediction and entropy of printed english. *Bell Systems Technical Journal*, pages 50–64, 1951.

[24] R. Volz, J. Kleb, and W. Mueller. Towards ontology-based disambiguation of geographical identifiers. In *I3*, 2007.