Bachelor Thesis Proposal

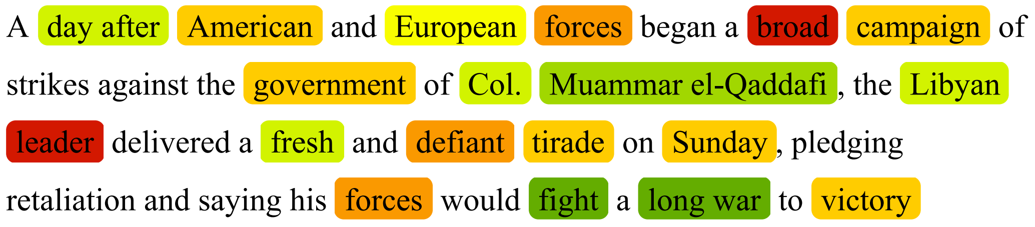
Joachim Daiber (4232126)

Abstract

DBPedia Spotlight[1] is a system for annotating DBPedia resources in plain text. By employing one of the core ideas of Linked Data, the representation of resources by URIs, the scope of the system goes beyond Wikipedia data alone and extends to the Web of Open Data. DBPedia Spotlight uses a model, which is based on the assumption that in a text, a DBPedia resource is likely to co-occur with a set of other words. To compute likely co-occurences of a DBPedia resource, article links in Wikipedia are interpreted as known mentions of DBPedia resources and their textual context is feed into the statistical model.

In the current version of DBPedia Spotlight, the annotation of plain text proceeds in three steps. In a first step, all possible mentions in the text are found by computing all substrings known to be possible textual representations of DBPedia resources. In the second step, the possible textual representations from the first step should be narrowed down to a list of plausible candidates. In the final step, all candidates are disambiguated in order to connect them to specific DBPedia resources. The second step involves an initial selection of candidates and should eliminate a number of mentions before they are even disambiguated. This step is currently included in the disambiguation step, and all candidates for which the training data includes a mapping to a DBPedia resource are considered. The scope of this thesis is to extend the candidate selection step to also be able to to handle common words and to evaluate the original system against the modified system and against other, similar systems.

The following example shows annotated mentions for a sentence from a newspaper article[2]. The colour of the annotation indicates the system's confidence (from green to red: high to low confidence).



A day after American and European forces began a broad campaign of strikes against the government of Col. Muammar el-Qaddafi, the Libyan leader delivered a fresh and defiant tirade on Sunday, pledging retaliation and saying his forces would fight a long war to victory

Figure 1: Example annotations by DBPedia Spotlight (colour indicates confidence)

There are several instances of common words or common word combinations for which the disambiguation step would be unnecessary and too expensive. These include annotations of verbs (*fight*[3]), common adjectives (*broad*[4], *fresh*[5], *defiant*[6]), combinations of words (e.g. *day after*[7]) and common nouns (e.g. *leader*[8], *victory*[9]). This example further demonstrates that it is

[1] Mendes, Pablo; Jakob, Max; García-Silva, Andrés; Bizer, Chris. "DBpedia Spotlight" <http://dbpedia.org/spotlight>

[2] http://www.nytimes.com/2011/03/21/world/africa/21libya.html?_r=1&hp

[3] Annotation: http://dbpedia.org/resource/Combat (Confidence > 0.8)

[4] Annotation: http://dbpedia.org/resource/Chris_Broad (Confidence > 0.1)

[5] Annotation: http://dbpedia.org/resource/Fresh_Air (Confidence > 0.3)

[6] Annotation: http://dbpedia.org/resource/Defiant_%28Star_Trek:_Deep_Space_Nine%29 (Confidence > 0.3)

[7] Annotation: http://dbpedia.org/resource/The_Day_After (Confidence > 0.6)

[8] Annotation: http://dbpedia.org/resource/Leadership (Confidence > 0.1)

not sufficient to use a list of stop words which would not be annotated, since, for example, the annotation "day after", which erroneously refers to the movie *The Day After* would be valid in other contexts (e.g. "Yesterday, I watched the movie Day After."[10].) This step further has the constraint that it needs to be fast, since it is only a minor step in the overall annotation process.

Hence, the aim of this thesis is to implement this selection step. For this, I plan to evaluate and combine a series of features and methods:

1. Linguistic information provided by a part-of-speech tagger[11] to detect non-ambiguous common verbs, adjectives and word combinations in the annotations
2. Information from various sources to detect common usages of nouns and compounds as opposed to specific mentions (cf. "day after" above):
    a. WordNet[12]
    b. Wiktionary links in Wikipedia articles
    c. Data from extensive linguistic corpora[13] for co-occurrence analysis

These features would then be combined into a classifier that would provide a fast decision if a mention is a possible candidate for a DBPedia resource and should be further disambiguated or if it should be ignored.

The modifications to the system would then be evaluated by comparing annotations by the original and modified system, by comparing results from the modified system to a set of gold standard annotations and to annotations from other, similar, systems. The work on the system will be done in collaboration with the DBPedia Spotlight team and the resulting extension will be incorporated into DBPedia Spotlight and shared as open source.

[9] Annotation: http://dbpedia.org/resource/Victory (Confidence > 0.4)

[10] Please note, that the training of DBPedia Spotlight can be both case-sensitive and case-insensitive and that the version currently deployed uses case-insensitive data.

[11] LingPipe <http://alias-i.com/lingpipe/>

[12] Princeton University "About WordNet." WordNet. Princeton University. 2011. <http://wordnet.princeton.edu>

[13] Quasthoff, U.; Richter, M.; Biemann, C. Leipzig Corpora Collection. <http://corpora.informatik.uni-leipzig.de/download.html>

Outline

1. Introduction
2. Problem outline
   - 2.1. Brief overview of the relevant parts of DBPedia Spotlight
   - 2.2. Description and relevance of the problem
   - 2.3. Classification of instances of the problem and examples from manual annotation
3. Relevant features and methods for detecting common mentions
   - 3.1. Overview
     - 3.1.1. Part-of-speech tagging
     - 3.1.2. Linguistic resources
       - 3.1.2.1. Co-occurrence data
       - 3.1.2.2. WordNet
       - 3.1.2.3. Wiktionary
   - 3.2. Combination of features and implementation
4. Evaluation of entity extraction systems
   - 4.1. Overview of evaluation techniques
   - 4.2. Golden standard evaluation with manual annotations
   - 4.3. Automatic evaluation against existing systems
     - 4.3.1. Scope of automatic evaluation
     - 4.3.2. Implementation of wrapper classes for existing systems
   - 4.4. Results of the evaluation
5. Summary and Conclusion