Who is it? Context sensitive named entity and instance recognition by means of *Wikipedia*

Ulli Waltinger and Alexander Mehler
Text Technology - Bielefeld University
Universitätsstrasse 15, 33602 Bielefeld, Germany
{Ulli_Marc.Waltinger, Alexander.Mehler}@uni-bielefeld.de

Abstract

This paper presents an approach for predicting context sensitive entities exemplified in the domain of person names. Our approach is based on building a weighted context but also a weighted people graph and predicting the context entity by extracting the best fitting sub graph using a spreading activation technique. The results of the experiments show a quite promising F-Measure of 0.99.

1. Introduction

Information extraction, especially the sub task in identifying entities in text is becoming more and more important. Tagging a token as a named entity and classifying them into predefined categories like organization or person and so on has been done quite successfully [5] [9]. See [1] for a current overview of named entity recognition tasks. Predicting the proper instance of a name within its context assumes to us therefore the next step to take. As a preassumption we are not focusing on the preprocessing of texts. In the preprocessing architecture [12] we have implemented a trigram HMM-Tagger following [4] with a F-measure of 0.975 trained and evaluated on the German Negra-Corpus [11]. The lemmatization module consists of a rule-based noun lemmatizer and a word form lexicon of around 4.9 million word-lemma pairs with a F-measure of 0.920. Named entity recognition is done by a simple rule-based module adopted from [6]. Therefore we are focusing in this paper on the next step in predicting the right instance of an entity classified as a person.

2. Who is it?

The topic of this paper is to tackle the task of disambiguation of named entities. In special we want to identify the proper instance of an ambiguous name. The idea of our approach is motivated by the game "*Who is it?*" where blindfolded participants trying to predict the name of a person by asking randomly questions and getting only a boolean as an answer. For example, if someone talks about: *boxing, ukraine, world champion, younger brother* one might be able to link this information to the name of *Wladimir Klitschko*. If we replace now the term *younger brother* with the name *Klitschko* there is a 0.5 chance in predicting the right instance, because we have two boxing brothers *Vitali* and *Wladimir Klitschko*. Since *Klitschko* is not a widespread name, our chances are quite good in picking the right instance. Considering just knowing the name *Müller* our chances of picking the right name instance out of a people database (see Section 3.1) goes down to 0.0029. This clearly shows that a semantic disambiguation of named entities is not a trivial task.

3. Algorithm of Prediction

An important precondition of our experiments is that any instance of a person's name appears within a textual environment or context (e.g. a section or paragraph of a text). The information that someone talks about *Helmut Kohl (politics)* and not about *Helmut Kohl (referee)* is only conveyed if the speaker gives her audience some background information as, for example, by using the terms *chancellor* or *party*. Adopting this idea to the domain of text we assume that the context of an instance is the lexical neighborhood of its name in the corresponding text. This context is delimited by units of the logical document structure of a text including paragraph, sentence etc. [10]. Therefore, if we look at an entity we have to incorporate its context by introducing a context window within the document structure. Once having set this context window, all included tokens than can be seen as a context instance of the examined entity by pointing to it.

3.1. Building a People Graph

Since we do not want to generate new entities [7, 8] but detecting the instance of a name, we can only be as successful as we already know at least one instance of this name. In our approach we utilize the *Wikipedia* as such a resource. By crawling the *Wikipedia-Category*:*People* we generated a database of $183,554$ different articles about people. In a second step, we extracted all hyperlinks which occur in one of these articles and defined them as a context link to the title of the article, the other content is dropped. Third, we have to convert the remaining tokens into a graph representation. We define a directed graph $G_{\text{people}} = (V, E, \omega)$, called *people graph*, with the set V of vertices and $E \subseteq V^2$ of edges. In our case, $V = P_1 \cup P_2$ consists of two subsets: the set P_1 of article names and the set P_2 of anchor names of those hyperlinks which start from articles in P_1. Further, $(v, w) \in E$ iff there exists an article named $w \in P_1$ such that $v \in P_2$ is the anchor name of a hyperlink in this article. Finally, $\omega\colon E \rightarrow \mathbb{R}_0^+$ is the edge weighting function. Note, that G_{people} is a bipartite digraph without multiple edges. Having done this, we have to care about the edge weighting function ω. In this regard we distinguish three different content structures of articles of persons in Wikipedia. The most significant information of a person as, e.g., her full name, birthday, death, profession or residence is most likely given in the first paragraph of the person's article. Thus, anchor names used as vertices of V are specially treated by ω. This also holds for the full or nickname if occurring in the first paragraph. More specifically, we define three classes of vertices in P_2:

- Class C_1 is the set of all vertices in P_2 for which there is at least one article in which they occur in the first paragraph as a substring of the title.
- Class C_2 is the set of all vertices in P_2 for which there is at least one article in which they occur in the first paragraph, but *not* as a substring of the title.
- Class C_3 is simply the set all remaining vertices, that is, $C_3 = P_2 \setminus (C_1 \cup C_2)$.

The next step of our algorithm is to weight each edge $e \in E$ by a function of the partition of P_2 into C_1, C_2, C_3. This is done by calculating the conditional probability $P(v|w)$ that $v \in P_1$ is the (person) name of an article in which $w \in P_2$ is occurring as the anchor of a link. In order to compute $P(v|w)$ we need to have frequency information about the occurrences of $w \in A_2$. Let, $f(w)$ be the total frequency of tokens of type w in all articles of our corpus. Then we estimate:

$$P(v|w) = \frac{f(w, v)}{f(w)} \tag{1}$$

where $f(w, v)$ is the frequency of w in the article named v. Finally, we define the weighting function ω for any edge $(w, v) \in E$ as follows:

$$\omega((w, v)) = \begin{cases} 1 \cdot \mathrm{idf}(w) \cdot P(v|w) : w \in C_3 \\ 2 \cdot \mathrm{idf}(w) \cdot P(v|w) : w \in C_2 \\ 3 \cdot \mathrm{idf}(w) \cdot P(v|w) : w \in C_1 \end{cases} \in [0, 3] \tag{2}$$

3.2. Building Context Clouds

In order to generate token related candidates, a lexical resources has to be build. The concept, in building a context cloud around an input token, is based on the exploiting of the document structure. In special we are using a pre-defined sentence window around the unknown token, therefore the technique of sentence-based statistical co-occurrence is adopted. The repeated occurrence of two words within a defined unit of information is called a statistical co-occurrence. As an adequate co-occurrence measure, we use the significance measure similar to the log-likelihood [2]. The significant co-occurrences reflect in this case a relation between two words and can be used for generating related terms for a given input token. The calculations in building the co-occurrence network are computed by *TinyCC* [3] on the basis of a lemmatized reference corpus (of 688,728 lemmata) extracted from the German newspaper *Die Zeit*. The significance measurement (see Figure 4) can be computed by the following algorithm. We define *k* as the number of sentences containing word *a* and *b* together. *ab* (see Figure 3) is (number of sentences with *a*)*(number of sentences with *b*) and *n* is total number of sentences in corpus.

$$x = \frac{ab}{n} \tag{3}$$

$$\sigma(a, b) = x - k * \log x + \log k \tag{4}$$

By means of this function we get a co-occurrence graph $G_\sigma = (V', E', \sigma)$ where V' is the set of all lemmata of our reference corpus and $E' \subseteq V'^2$ the corresponding set of edges. Note, that a global threshold τ is introduced after sorting E' by means of σ in descending order and keeping only those edges (v, w) in E' for which $\sigma(v, w) > \tau$. This is done in order to reduce the number of edges. Suppose now we have a given text x in which we observe an ambiguous name, say $v \in A_1$. Based on the textual context of v in x we build a so called context cloud which is a graph (V'', E'', σ') such that $V'' \subseteq V'$ is the set of all lemmata of the co-occurrence graph G_σ occurring in x and E'' and σ' are the restrictions of E' and σ' to V'', respectively. In other words the context graph of a name is a subgraph of the co-occurrence graph. Having this, we are able to send a request for an input word $w1$ and getting as an response a set of nodes ranked by ω representing the context instances $w1_{\text{i}}$ of $w1$.

3.3. Predicting names

In predicting the right instance defined as L_{instance} we conduct a spreading activation technique. The algorithm works as follows (see Algorithm 1). We assign for each label $v \in A_1$ - these are those v_{i} who are not pointing to an other vertex in E (our entity instances) - of G_{people} to its own activity class C_{i}. The value of all activity classes W_{i} are set to zero. For each input token t, within the context window, we build the context cloud w as described in the previous section. For each generated context term wi we than add the edge value ω to those $v \in A_1$ classes, where an edge between V and V' exists. Note z defines the number of context terms used for prediction. The new activation value W_{i} of a class C_{i} is calculated by building the sum of the edge weight ω_{people} of the people graph and the edge weight σ' of the co-occurrence graph. See Table 1 for an example of the activity ranking in our experiments. By that, for every new entered term, the responding $v \in A_1$ classes will grow in their activation value. This will automatically build a ranking of $v \in A_1$ sorted by their activation value. After the entire context window z is computed, we pick this entity instance L_{instance}, whose $v \in A_1$ value was maximized during the process (***Figure** 6*). This simple algorithm is yet very effective and low in complexity (cost of lookup: $\log(w)$). Classes will stabilize during the process, because the activation values will be multiple assigned to different classes. Therefore, non relevant $v \in A_1$ will grow only, if an input token is relevant to them. Thus, in the sum they will still have a smaller value than the selected one.

$$C_{\text{i}} = W_{\text{i}}(E_{\text{i}} \bigcap E'_{\text{i}}) \tag{5}$$

$$W_{\text{i}} = W_{\text{i-1}} + \omega_{\text{people}} + \sigma' \tag{6}$$

$$L_{\text{instance}} = \arg \max_{0<w<z+1} \{C_{\text{i}}\} \tag{7}$$

4. Experiment

The conducted experiment was split into four parts by varying the foreknown knowledge. In the first run, we assumed to have both the first and the surname given. This is in practice the most common case to handle. In a second run, we set only the surname as a prerequisite. Third, we set only the forename as our foreknown knowledge. In the last scenario, we entered the algorithm without any cognition but its context. The evaluation corpus was build by two volunteers, collecting newspaper articles of randomly chosen persons. They were prompted to copy a snippet - the paragraph a name was mentioned - of a random article about the person and removing all occurrences of the name in the paragraph. Therefore, there was no chance to have a detailed link to the predicting name. Since we wanted to have a real life scenario, we told the volunteers not to analyze the paragraph on appropriate descriptive terms, but more or less blind folded copying the paragraph into the corpus file. As a result, we had a corpus size of 195 unique person names, represented respectively with one paragraph. After that, we interlinked each paragraph with the actually meant name instance of our people database by analyzing the entire article and the different possible instances of the name.

Algorithm 1 Predicting names.

```
set all activation values of all $v \in A_1$ to $0$
for each token $t$ of the input text do
    build context cloud $w$ of $t$
    for each item in $w$ do
        if $w$ is element of $A_2$ then
            for augment for all $v \in A_1$ for which $(w, v) \in E$ do
                $v = v + activation\_value.$
            end for
        end if
    end for
end for
Select that $v \in A_1$ which has the highest activation value.
```

5. Evaluation

For the evaluation we applied the standard information retrieval metric (F1-Measure) to assess our results. The results of the entire experiment, implemented in our preprocessing architecture, are presented in Table 2. We conducted a parameter study by diversifying the activity value W_{i} as defined in Section 3.3. In the first place we set $\omega_{\text{peo\_w}} + \omega_{\text{sig\_w}} = 1$ (see Table 1 (e1)). Second, we used $\omega_{\text{peo\_w}}$ and $\omega_{\text{sig\_w}}$ without their *idf* (see Table 1 (e2)) and third with their *idf* value (see Table 1 (idf)). The outcome of our evaluation on the first experiment shows, that if a full name is given we are able to predict with an F-Measure of 0.99 the right instance. Nevertheless, the more interesting part is the conducted second experiment - only the surname is given. In this case, we were still retrieving an F-Measure of 0.84 though we have an increase in selected possibilities.

The third experiment reaches a measure of 0.71, knowing only the first name leave a lot of prospect in predicting the right instance. Confronting our application with no default knowledge, just a plain paragraph with no name to interlink to, let us still retrieve an F-Measure of 0.04, which is not sufficient but also not surprising. In summary, we achieved a very satisfying basis for the disambiguation for

Searched Name	Instances	Activity Ranking
H. S. (athlete)	H. S. (athlete)	2.4
	H. S.(skier)	0.09
M. K. (writer)	M. K. (writer)	0.024
	M. K.(politics)	0.0005
	M. K.(soccer)	0.00004

Table 1. Activity Ranking of Hubert Schwarz (H. S.) and Michael Krüger (M. K.)

Knowledge	F-Measure
Fullname (idf)	0.99
Surname (idf)	0.84
Forename (idf)	0.71
-	0.04
Fullname (e2)	0.98
Surname (e2)	0.82
Forename (e2)	0.62
Fullname (e1)	0.96
Surname (e1)	0.81
Forename (e1)	0.72

Table 2. Evaluation Results

predicting the right name instance when having at least one constituent of a name set. Since we assigned the context window to one paragraph only, the results of the fourth experiment (no foreknown knowledge) are not surprising.

Using this technique for an other domain like product names [*Canon: is it an EOS or an Snapshot*] or city name [*Berlin: in Germany or in the USA*], is easy to adopt. In our future work, we will focus on exactly those problems.

6. Conclusion

This paper presents an approach for predicting context sensitive named entities exemplified on the domain of people names. Doing this we assume that an instance of a name appears always with in its context in a document structure. Therefore the method exploits the context-surrounding of a token, by analyzing the border-sentences of an entity. The information of the incorporated context was then used for building an expanded context graph around the unknown entity. This was done by querying a co-occurrence network, keeping the most significant edges to the context-instance. Our evaluation scenario was split into four different tasks, by consequently reducing the foreknown knowledge about the entity. The F-measures of 0.99 rsp. 0.98 are quite promising.

Acknowledgement

Financial support of the German Research Foundation (DFG) through the *Excellence Cluster 277 Cognitive Interaction Technology* (KnowCIT), the SFB 673 *Alignment in Communication* (X1), the Research Group 437 *Text Technological Information Modeling* (A4) and *Scientific Library Services and Information Systems (LIS)* at Bielefeld University is gratefully acknowledged.

References

[1] Nist 2007 automatic content extraction evaluation official results, 2007.

[2] C. Biemann, S. Bordag, G. Heyer, U. Quasthoff, and C. Wolff. Language-independent methods for compiling monolingual lexical data. In *CICLing*, pages 217–228, 2004.

[3] C. Biemann, U. Quasthoff, G. Heyer, and F. Holz. ASV Toolbox – A Modular Collection of Language Exploration Tools. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC) 2008*, 2008.

[4] T. Brants. Tnt - a statistical part-of-speech tagger. In *Proceedings of theSixth Applied Natural Language Processing Conference ANLP-2000, Seattle, WA.*, 2000.

[5] S. Cucerzan. 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)*, pages 708–716, 2007.

[6] H. Cunningham, K. Bontcheva, V. Tablan, and D. Maynard. Gate - a general architecture for text engineering, 2007.

[7] G. Friedrich and K. Shchekotykhin. Nameit: Extraction of product names. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference, 2006*. IEEE, 2006.

[8] M. Jimnez. Generation of named entities. In *MT Summit VIII. Santiago de Compostela, Spain, 2001*. European Association for Machine Translation, 2001.

[9] J. Kazama and K. Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707, 2007.

[10] R. Power, D. Scott, and N. Bouayad-Agha. Document Structure. *Computational Linguistics*, 29(2):211–260, 2003.

[11] H. Uszkoreit, T. Brants, S. Brants, and C. Foeldesi. Negra corpus, 2006.

[12] U. Waltinger and A. Mehler. Web as preprocessed corpus: Building large annotated corpora from heterogeneous web document data. In preparation, 2008.