DFKI2: An Information Extraction Based Approach to People Disambiguation

Andrea Heyl
German Research Center for
Artificial Intelligence - DFKI,
Saarbrücken, Germany
andrea.heyl@dfki.de

Günter Neumann
German Research Center for
Artificial Intelligence - DFKI,
Saarbrücken, Germany
guenter.neumann@dfki.de

Abstract

We propose an IE based approach to people disambiguation. We assume the mentioning of NEs and the relational context of a person in the text to be important discriminating features in order to distinguish different people sharing a name.

1 Introduction

In this paper, we propose a system with a linguistic view on people disambiguation that exploits the relational and NE context of a person name as discriminating features.

Texts about different people differ from each other by the names of persons, places and organizations connected to these people and by the relations in which a person's name is connected to other entities. Therefore we had the hypothesis that the NEs in the documents for a person name should be a main distinctive criterion for disambiguating people.

Furthermore, the relational context of a person name should also be able to give good clues for disambiguation. Sentence patterns related to a name, i.e. patterns that contain the name as subject or object like "be(Person X, lawyer)" often convey uniquely identifying information about a person.

Our system was not built specifically for the web people search task WePS (Artiles et al., 2007), but is an early version of an IE system that has the more general goal to discover relations between NEs. We see the WePS task as a specific instance of the set of tasks our system should be able to handle. Therefore, we only adapted it slightly to work with the WePS data, but did not make any further customization w.r.t. the special requirements of people disambiguation. As our system was built to handle pure texts rather than structured web pages, we relied completely on linguistic information and did not exploit the html structure of the documents provided.

2 Related Work

Our system was inspired by the preemptive and on-demand IE approaches by Sekine and Shinyama (Sekine, 2006; Shinyama, 2006) that cluster newspaper articles into classes of articles that talk about the same type of event. They proposed a system to discover in advance all possible relations and to return them in form of tables.

We took the idea of distinctive personal attributes as a criterion for disambiguation from the work of Bollegala et al. (2006). They propose an unsupervised learning approach to extract phrases that uniquely identify a person from the web and use these discriminative features for clustering.

3 System Overview

The goal of the WePS task is to cluster the top 100 web pages returned by a web search engine for a certain name as search query and classify them w.r.t. the underlying different people they refer to.

The problem of clustering documents about people into different entities can be seen as two subproblems: The determination of the correct number of clusters and the clustering of the given documents into this number of entities. These problems could either be solved consecutively by first estimating the number of classes and then produce this pre-

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 137–140,
Prague, June 2007. ©2007 Association for Computational Linguistics

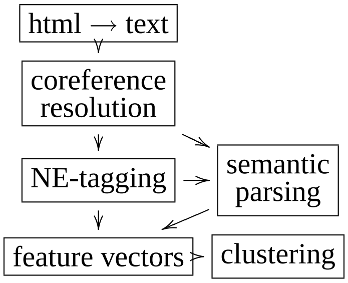

Figure 1: System Overview

set number of clusters or by determining the number of classes dynamically during the clustering process.

Figure 1 gives an overview of our system, that clusters web documents into a pre-defined number of classes, thereby being only concerned with the second problem and neglecting the estimation of different namesakes for now.

Every web page in the WePS training data is represented by the set of its files. As our system works on plain text only, we first needed to separate the textual parts of all files. Therefore, we extracted the text from the html pages. We merged the texts from all different html pages belonging to a single website into one document so that we obtained for every person's name 100 text files as the basis for further clustering.

These text files were processed by a coreference resolution tool. On the resulting texts, we ran both an NE tagger and an NLP tool for semantic parsing. This tool represents sentences containing the respective person name as predicate argument structures.

We constructed two feature vectors for each file based on the counts of the NEs and predicate argument structures that contain the specific person name. Those feature vectors were our basis for the clustering process.

The clustering unit of the system consecutively merged clusters, that at first contained a single file each, until the pre-set number of classes was reached and returned the clustering as an xml file.

4 System Components

4.1 Estimating the Number of Classes

In principle, the number of different people that are represented in the data cannot be known in advance. However, for the clustering process, either the number of classes has to be fixed before clustering, or some kind of termination criterion has to be found that tells the algorithm when to stop clustering.

A good estimation of the number of different entities is a necessary prerequisite for successful clustering. Clustering into too many classes would mean assigning documents to classes that have actually no own entity they refer to. Clustering into too few classes means merging two entities into one class.

Our initial intuition was to distinguish people by normally unique properties, like phone numbers or email addresses. So we assumed that the number of different email addresses and phone numbers occurring in all documents for one name would be a good means to estimate the number of different persons sharing this name, but we could not find any correlation between these features and the class number.

Therefore, we decided to estimate the average number of classes from the training data. The average number of different people for one name in the training data was about 18. Based on the observation that an underestimated number of classes leads to better results than assuming too many classes, we decided to guess 12 different persons for each name.

4.2 Preprocessing

For the extraction of plain text information from the web pages, we used the html2text [1] converter. In case that a web page consisted of more than one html document, we put all the output from the converter into one single file. By omitting any wrapping of the html pages, we obviously lost useful structural information but got the textual information for our linguistic analysis.

Afterward, we applied several linguistic preprocessing tools. We used coreference resolution to replace pronouns referring to a person, and variations of a name (like "Mr. Smith" after a mention of "John Smith" earlier in the text) with the person's name in the form of its first mention in the text.

For NE-tagging, we used the three NE types PERSON, LOCATION and ORGANIZATION. For both NE tagging and coreference resolution, we used the LingPipe toolkit [2]. We counted the occurrences of every NE in every file and replaced all instances by their specific NE type combined with a uniquely

[1] http://www.mbayer.de/html2text/index.shtml

[2] http://www.alias-i.com/lingpipe/

identifying number, e.g. we replaced all occurrences of “Paris” with “LOCATION27”, in order to ensure that the predicate argument parser could work correctly and would not split up multi-word NEs into two or more arguments.

We passed all sentences with NEs that contained the specified persons family name (e.g. “Mr. Cooper” for the name “Alvin Cooper”) to MontyLingua [3], that returns a semantic representation of the sentence like (“live” “PERSON2” “in LOCATION3”). These representations abstract from the actual surface form of a sentence as they represent every sentence in its underlying semantic form (“predicate” “semantic subject” “semantic object1”...) rather than just determining the syntactic subject and objects of a sentence. We called these structures “patterns” and kept only those that actually contained the respective NE.

4.3 Clustering

We decided on building two vectors for every text file, one for the NEs and one for sentence patterns connected to a person’s name in order to give to the NEs a weight different from that for the patterns.

After tagging the documents for NEs, we counted the frequency of the different occurring NEs for one name. We built a first feature vector for each document that contained as entries the counts of the occurring NEs in this document. We set a threshold n to use only the n best NEs in the vectors, counted over all documents for one name. We then built for every document a second feature vector containing the counts of the MontyLingua patterns for the document.

For the actual clustering process, we used hierarchical clustering. We started with every file, represented by a pair of normalized feature vectors, constituting a single cluster. As distance measurement we used the weighted sum of the absolute distances between the centers of two clusters with regard to both feature vectors, respectively, i.e. we chose distance $= w \cdot \text{distance}_{\mathit{NEs}} + \text{distance}_{\mathit{patterns}}$. In every step, we made a pairwise comparison of all clusters and merged those with the lowest distance. The clustering terminated when the algorithm came down to the pre-set number of 12 clusters. So far we have not made any further use of the binary tree structure within each cluster.

We assigned every file to exactly one cluster. We had neither a “discarded” category nor did we handle the possibility that a page refers to more than one person and would hence belong to different clusters.

5 Experiments

5.1 Training of Parameters

We evaluated the system on the provided WePS training data to estimate the following parameters: number of classes, number of best NEs to be considered and weight of the NE vector compared to the pattern vector.

The relevant evaluation score is the F-measure ($\alpha = 0.5$) as the harmonic mean of purity and inverse purity as described by Hotho et al. (2003).

As our attempt to use distinctive features for the estimation of class numbers failed, we examined the influence of a wrongly estimated number of classes on the clustering results. Table 1 shows exemplarily for 2 person names how the F-measure varies if the correct number of classes is incorrectly assumed as a higher or lower value. We concluded that it is better to estimate the class number too low than too high.

name	A. Macomb	E. Fox
correct number of classes	21	16
10 classes assumed	0.76	0.80
12 classes assumed	0.75	0.75
14 classes assumed	0.72	0.76
16 classes assumed	0.69	0.60
18 classes assumed	0.60	0.58
20 classes assumed	0.48	0.72
22 classes assumed	0.56	0.55
24 classes assumed	0.59	0.58
26 classes assumed	0,52	0.56

Table 1: F-measure for different numbers of assumed classes

Primarily meant as a means to reduce computation time, we gave our system the possibility not to use all occurring NEs for clustering, but only a certain number of entities with maximal frequencies. Test runs did not confirm our hypothesis that considering a higher number of NEs leads to better results (cf. table 2). For both training of the number of NEs and the NE weight we assumed that we already knew the correct class number.

As the F-measure did not increase for more considered NEs, we believe that the most important NEs

[3] http://web.media.mit.edu/ hugo/montylingua/

are already covered within the best 100 and that adding more NEs rather adds coincidental information than any new important facts. Usually, the best 100 NEs already cover most of those which occur more than once in a text.

NEs	average. F-measure
100	0.66
200	0.68
500	0.68
1000	0.67

w	average F-measure
0.5	0.66
1.0	0.68
2.0	0.68
4.0	0.67

Table 2: varying the number of considered entities and weight of the feature vectors

The third parameter to estimate was the weight w given to the NE feature vector compared to the feature vector for sentence patterns. During training, this weight also appeared to have little influence on the clustering results (cf. 2). We have the hypothesis that sentence pattern detection is not very successful for the often unstructured web page texts.

5.2 Results for WePS Test Data

In the WePS evaluation, our system scored with a purity of 0.39, an inverse purity of 0.83 and a resulting overall F-measure ($\alpha = 0.5$) of 0.5.

One main reason for our test results to be worse than our training results is the fact that the test data had a much higher average number of classes (about 46 classes). Our F-measure was best for those names with the fewest number of referents. We had an average F-Measure ($\alpha = 0.5$) of 0.66 for those names with less than 30 instances compared to an overall average of 0.50. These numbers show the importance of a correct estimation of the assumed number of referents for a name.

Our purity was much lower than the inverse purity, i.e. there is too much noise in our clustering compared to the real partition, whereas the real clusters are well covered by our clustering. This is due to a too low estimation of the number of referents.

6 Conclusions and Future Work

One obvious improvement , that would accommodate the general relation extraction idea of our system, is to include the use of structural information from the html documents in addition to our purely linguistic view on web pages. Additionally, we should weight our NEs using e.g. a TF/IDF formula.

A promising direction for further research in people search will certainly include a better control of the number of classes. This could be done either by estimating this number in advance, or by setting the number of classes dynamically during clustering. The latter could include comparing the size of the current clusters to the overall feature space of all clusters or an approach of counting occurrences of uniquely identifying attributes within a cluster.

This second approach could match the original purpose of our system, namely to build tables that represent the most salient relations in a set of documents in the way Sekine and Shinyama did. If such a table, that represents the slots of a relation in its columns and every article in a row, is built for all documents in a cluster, we would expect the table to contain roughly the same information in every row. One could define a consistency measure for the resulting tables and stop clustering as soon as the tables are no longer consistent enough, i.e. when they contain too much contradictory information.

Acknowledgment

The work presented here was supported by a research grant from the Investitionsbank Berlin to the DFKI project IDEX (Interactive Dynamic IE).

References

Javier Artiles, Julio Gonzalo and Satoshi Sekine. 2007. *The SemEval-2007 WePS Evaluation: Establishing a Benchmark for the Web People Search Task.* Proceedings of Semeval 2007, ACL.

Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka. 2006. *Extracting Key Phrases to Disambiguate Personal Name Queries in Web Search.* Proceedings of the Workshop on How Can Computational Linguistics Improve Information Retrieval, p. 17–24.

Andreas Hotho, Steffen Staab and Gerd Stumme. 2003. *Wordnet Improves Text Document Clustering.* Proceedings of the Semantic Web Workshop at SIGIR-2003, 26th Annual International ACM SIGIR Conference, Toronto, Canada.

Satoshi Sekine. 2006. *On-Demand IE.* International Committee on Comp. Ling. and the ACL.

Yusuke Shinyama and Satoshi Sekine. 2006. *Preemptive Information Extraction using Unrestricted Relation Discovery.* Human Language Technology conference - North American chapter of the ACL annual meeting; New York City.