Fuzzy Ants Clustering for Web People Search

Els Lefever [*] [1,2], Timur Fayruzov[2], Veronique Hoste[1,2], Martine De Cock[2,3]

[1]LT3, Language and Translation Technology Team, University College Ghent
Groot-Brittanniëlaan 45, 9000 Gent, Belgium
[2]Dpt. of Applied Mathematics and Computer Science, Ghent University
Krijgslaan 281 (S9), 9000 Gent, Belgium
[3]Institute of Technology, University of Washington, Tacoma
1900 Commerce Street, Tacoma WA–98402, USA

ABSTRACT

A search engine query for a person's name often brings up web pages corresponding to several people who share the same name. The Web People Search (WePS) problem involves organizing such search results for an ambiguous name query in meaningful clusters, that group together all web pages corresponding to one single individual. A particularly challenging aspect of this task is that it is in general not known beforehand how many clusters to expect. In this paper we therefore propose the use of a Fuzzy Ants clustering algorithm that does not rely on prior knowledge of the number of clusters that need to be found in the data. An evaluation on benchmark data sets from SemEval's WePS1 and WePS2 competitions shows that the resulting system is competitive with the agglomerative clustering Agnes algorithm. This is particularly interesting as the latter involves manual setting of a similarity threshold (or estimating the number of clusters in advance) while the former does not.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*

General Terms

Algorithms, Experimentation, Measurement

Keywords

Web People Search, Web People Disambiguation, Fuzzy Ants Clustering, hierarchical clustering, document clustering, WePS

1. INTRODUCTION

Searching for people on the web is a very popular online activity that is receiving increasing support from specialized search engines. Besides the major search engine Yahoo! offering a people search service[1], there is a fair amount of smaller search engines whose core business is specifically web people search, such as pipl[2] and spock[3]. One significant problem faced by all of these "web people finders" is person name ambiguity, i.e. the fact that one person name can refer to different individuals. In the context of this paper, Web People Search (WePS) refers to the problem of organizing all search results for a person name query into meaningful clusters that group together all web pages corresponding to one single individual.

Given the high relevance of the WePS problem for various natural language processing domains such as information extraction, cross-document summarization and question answering, a first WePS task was organized in the framework of SemEval 2007[4], an international competition on semantic evaluation, which was held in conjunction with ACL-2007. The success of this first WePS competition (16 participating systems) has lead to a follow-up competition (WePS II) which was organized in conjunction with WWW2009. This paper describes the results of a system that we developed to participate in this competition.

A particularly challenging aspect of the WePS problem is that it is usually not know beforehand how many clusters to expect in the search results for a person name query. Some unusual names occur rarely, while other, more common names, are shared by a large group of people. Furthermore, celebrity names tend to monopolize search results. For instance, while there may be many people out there with a not very unusual name like Michael Jordan, the first 100 results returned by a search engine might be for a large part about the American basketball player and about the computer science professor at UC Berkeley. This makes it hard to predict how many different individuals will be covered in the first search results, which is problematic since most clustering algorithms require an estimate of the number of clusters that needs to be found in the data.

In this paper we propose the use of a Fuzzy Ants clustering algorithm that does not rely on such prior knowledge. Ant based clustering algorithms are inspired by the clustering of dead nestmates, as observed with several ant species under laboratory conditions. Without negotiating about where to gather the corpses, ants manage to cluster all corpses into one or two piles. The conceptual simplicity of this phenomenon together with the lack of centralized control and the lack of a need for a priori information are the main motivations for clustering algorithms inspired by this behavior.

[*]els.lefever@hogent.be (corresponding author)
[1]http://people.yahoo.com/

WWW2009, April 20-24, 2009, Madrid, Spain.

[2]http://www.pipl.com/
[3]http://www.spock.com/
[4]http://www.senseval.org/

Real ants are, because of their very limited brain capacity, often assumed to reason only by means of rules of thumb. The Fuzzy Ants clustering method proposed in [27] is inspired by this observation: in this approach the desired behavior of artificial ants and, more precisely, their stimuli for picking up and dropping items is expressed flexibly by fuzzy IF–THEN rules. Like all ant-based clustering algorithms, no initial partitioning of the data is needed, nor should the number of clusters be known in advance. The machinery of approximate reasoning from fuzzy set theory endows the ants with some intelligence. As a result, on each time step the ants are capable of deciding for themselves whether to pick up or drop an item or a heap, and a clustering automatically emerges from this process.

The Fuzzy Ants algorithm has been successfully applied to group search results for ambiguous queries such as *rem*, *travelling to Java*, and *salsa* [27]. To this end, snippets returned by the search engine are turned into bags of words with a binary weighting scheme and subsequently clustered. While this method is useful for detecting documents that talk about the same topic, the WePS problem calls for a different approach. Indeed, various web pages about the same individual might contain a significantly different kind of content, such as a professional web page versus an account page on a social networking site. Furthermore, the presence of a biographical fact or a telephone number might give an important clue on the individual at hand. While such information is rather rare, it is very reliable: we can be almost sure that identical phone numbers in different documents refer to the same physical person. In our approach we therefore represent every web document as a rich feature vector, containing information about biographical facts, named entities, telephone and fax numbers, URL and email addresses, geographic location, IP location, as well as distinctive keywords from the title and the snippet, and a weighted bag of words extracted from the full text of the document.

The remainder of the paper is organized as follows: in Section 2 we give an overview of related research, while in Section 3 we discuss the construction of the feature vectors. The flow of the clustering algorithms is discussed in Section 4, and in Section 5 we elaborate on the experiments and results. Section 6 summarizes the main findings of the paper and brings up some ideas for future work.

2. RELATED RESEARCH

The Web People Search task has already been approached from different angles. A wide range of features (extracted from the web page content as well as from external data) and algorithms (different classification and clustering algorithms) have been explored.

The task has often been considered as a multi-document coreference task ([4],[11],[23]) where a system has to decide whether two instances of the same name occurring in different documents refer to the same individual or not. Bagga and Baldwin [4] perform coreference resolution within each document in order to form coreference chains. Next they produce a summary of each person within each document by taking the surrounding context of these reference chains and convert these summaries into a bag of words representation. These are then clustered, using the cosine distance to measure similarity.

Another direction leads to unsupervised clustering based on a rich feature space. Mann and Yarowsky [18], for example, extract features containing biographical facts, such as birth and death place or date, that are combined with associated names (such as family and employment relationships and nationality). To extract these patterns, they further develop the method of Ravichandran and Hovy [25] that bootstraps information extraction patterns from a set of example extractions. They finally combine learned patterns and hand-coded rules. In this way each instance of the ambiguous name is represented by a feature vector and clustering is done by grouping the most similar feature vectors, using a bottom-up centroid agglomerative clustering based on the cosine similarity distance.

Several semantic based approaches ([22],[5]) have been proposed as well. Pedersen [22] presents an unsupervised approach that uses statistically significant bigrams in the documents to be clustered ("Significant" meaning that the log-likelihood ratio between the two words is greater than a given threshold). A matrix based on these bigrams is built with the rows representing the first word, and the columns representing the second word in the bigram; each cell contains the log-likelihood ratio associated with the bigram. Because of its large size and sparsity, Singular Value Decomposition is applied to reduce the dimensionality. Afterwards, instances (documents) that have similar context vectors are placed into the same cluster.

Other researchers use additional web resources for better measuring document similarities ([20],[6],[13]). Vu et al. [20] use web directories such as Dmos: www.dmoz.org, Google: directory.google.com and Yahoo: dir.yahoo.com as an additional knowledge base. These collections of web documents are categorized according to different topics and can be used to enrich the extractable information in web documents. In this way, they determine the topic of the web document and link other documents containing similar contexts.

In previous work [16] we presented a hybrid approach that combined the results of both classification and clustering. First, supervised classification based on feature vectors containing binary and symbolic disambiguation information on pairs of documents is done by means of the eager RIPPER rule learner [7]. Second, different clustering approaches are applied on the weighted keyword matrices. In a final step, the "seed" clusters that are obtained by the classification algorithm, are enhanced by the results from the clustering algorithms.

Our present work is mostly related to the research of Mann and Yarowsky, in the sense that we also do unsupervised clustering on a rich feature space. It is different from the approaches mentioned above because of the fuzzy ants clustering algorithm we use, that is able to cluster without any kind of a priori information such as the required number of output clusters, and because of the integration of very different features (biographical facts, named entity overlap, geographic location information, URL, email, and telephone number overlap and weighted keywords) into one feature vector per document.

3. FEATURE CONSTRUCTION

For each document we first construct a rich feature vector that combines biographical facts and distinctive characteristics for a given person, two geographic features and a list of weighted keywords. The input for our feature extraction module is a collection of objects for every person name that

contains for each object a title, a snippet and the full html web document. The ultimate goal of our algorithm consists in clustering all documents that refer to the same individual.

All documents are preprocessed by means of a memory-based shallow parser (MBSP) [8] and the following preprocessing steps are taken. Tokenization (i.e. splitting punctuation from adjoining words) is performed by the MBSP by a rule-based system using regular expressions. Part-of-speech tagging and text chunking is performed by the memory-based tagger MBT [9], which was trained on text from the Wall Street Journal corpus in the Penn Treebank [19], the Brown corpus [15] and the Air Travel Information System (ATIS) corpus [12]. During text chunking syntactically related words are combined into non-overlapping phrases.

3.1 Person distinctive features

We extract a set of features from the content of the web pages that are characteristic for a given person:

- **biographical facts**
  Date of birth and death are retrieved from the web pages by means of regular expressions (e.g. *was born in, died in, death on, etc.*).

- **named entities**
  We extract three named entity features: named entities referring to the focus name of the given document set (e.g. *Ann Hill Carter Lee* and *Jo Ann Hill* for *Ann Hill*), location names and other named entities. All named entities are extracted by combining the syntactic information that is generated by the memory-based shallow parser and lookup in gazetteers containing person and location names.

- **Telephone and Fax numbers**
  Telephone and Fax numbers are extracted by means of regular expressions that are a combination of fixed digit patterns and digit strings that are preceded by a telephone/fax indicator (e.g. *Tel., Fax, telephone, nr., etc*). This information is very rare, but very reliable (we can almost be sure that identical numbers in different documents refer to the same physical person).

- **URL and email addresses** As we assume that overlapping URLs, parts of URLs (domain addresses) and email addresses point to the same individual, we also extract this information by a combination of pattern matching rules and HTML markup information (HTML *href* tag). The document link is added to the set of links that are extracted from the content of the web pages. Very short and common URLs (e.g. *index.html*) are filtered from the list.

3.2 Geographical features

Two geographical features are constructed for each document:

- **geographical location feature**
  The orthographic location strings that are identified during the named entity extraction are mapped to their geographical coordinates by using the GeoNames database[5]. These coordinates are used for measuring the distance between locations in document pairs.

- **IP location feature**
  For the IP location feature, we start from the simple hypothesis that if two documents are hosted in the same city (because they share the same IP prefix), they probably refer to the same person. To convert the IP addresses into city locations, we use the MaxMind GeoIP(tm) open source database[6].

3.3 Bag of weighted keywords

We select distinctive keywords from titles and snippets, as well as from the content of the web pages themselves. Content words from the titles and snippets (with exception of the target name itself) are supposed to be very informative. Therefore they are all stored and compared for each document pair: in case there is overlap, this binary feature is set to "1", in case there is no overlap it is set to "0".

Keywords that are selected from the content of the web pages are treated differently, as we only want to store keywords that are relevant for the target document. First, the web page is cleaned (markup information and other HTML information is removed), tokenized and Part-of-Speech tagged. Only content words (verbs, nouns and adjectives) are selected for further processing. In order to detect relevant keywords for each document, we compute the TF-IDF score (term frequency - inverse document frequency) for each term-document pair, i.e. the relative frequency of the term in the document compared to the frequency of the term in the entire document corpus [26].

4. CLUSTERING

For each pair of documents, a comparison vector is constructed that contains binary features that measure the overlap for highly informative but sparse features between the two documents (person distinctive features, IP address overlap and keywords from snippets and titles) and numeric features (geographical distance feature and cosine similarity between weighted keyword vectors).

The second step consists of aggregating the comparison vector into one value that belongs to the interval $[0, 1]$. The aggregation step is performed by taking a weighted average.

As information gain has a tendency to favor features with many possible values over features with fewer possible values, we used a normalized version of information gain, called **gain ratio** [24], as weighting metric.

The information gain of a feature i is calculated as follows. Assume we have C, the set of class labels (a binary set in our case: document pairs belong to the same cluster or not) and V_i, the set of feature values for feature i. With this information, we can calculate the database information entropy. The probabilities are estimated from the relative frequencies in the training set.

$$H(C) = -\sum_{c \in C} P(c) \log_2 P(c)$$

The information gain of feature i is then measured by calculating the difference in entropy between the situations with and without the information about the values of the feature:

$$w_i = H(C) - \sum_{v \in V_i} P(v) \times H(C|v)$$

[5]http://www.geonames.org/about.html

[6]http://www.maxmind.com/app/geolitecity

Gain ratio, is a normalized version of information gain. It is information gain divided by split info $si(i)$, the entropy of the feature values. This is just the entropy of the database restricted to a single feature.

$$w_i = \frac{H(C) - \sum_{v \in V_i} P(v) \times H(C|v)}{si(i)}$$

$$si(i) = -\sum_{v \in V_i} P(v) \log_2 P(v)$$

4.1 Fuzzy Ants Clustering

The first clustering algorithm we apply is the Fuzzy Ants clustering. Ant-based clustering algorithms are usually inspired by the way ants cluster dead nestmates into piles, without negotiating about where to gather the corpses. These algorithms are characterized by the lack of centralized control or a priori information (see e.g. [10], [17], [21]), which makes them very appropriate candidates for the task at hand. Since the Fuzzy ants algorithm does not need initial partitioning of the data or a predefined number of clusters, it is very well suited for the Web People Search task, where we do not know in advance how many clusters (or individuals) correspond to a particular document set (or person name in our case).

A detailed description of the algorithm is given by Schockaert et al.[27]. It is an extension of Monmarché's algorithm [21] that involves a pass in which ants can only pick up one item as well as a pass during which ants can only pick up an entire heap. In [27], a fuzzy ant-based clustering algorithm was introduced where the ants are endowed with a level of intelligence in the form of IF–THEN rules that allow them to do approximate reasoning. As a result, at any time the ants can decide for themselves whether to pick up a single item or an entire heap, which makes a separation of the clustering in different passes superfluous.

We have experimented with a different number of ants runs and fixed the number of runs to 800 000 for our experiments. In addition, we have also evaluated different values for the parameters that determine the probability that a document or heap of documents is picked up or dropped by the ants and kept following values for our experiments:

n1	probability of dropping one item	1
m1	probability of picking up one item	1
n2	probability of dropping an entire heap	5
m2	probability of picking up a heap	5

Table 1: Parameter settings for fuzzy clustering

4.2 Hierarchical Clustering

The second clustering algorithm we apply is an agglomerative hierarchical approach. This clustering algorithm builds a hierarchy of clusterings that can be represented as a tree (called a dendrogram) which has singleton clusters (individual documents) as leaves and a single cluster containing all documents as root. An agglomerative clustering algorithm builds this tree from the leaves to the top, in each step merging the two clusters with the largest similarity. Cutting the tree at a given height gives a clustering at a selected number of clusters. We have opted to cut the tree at different similarity thresholds between the document pairs, with intervals of 0.1 (e.g. for threshold 0.2 all document pairs with similarities above 0.2 are clustered together).

For our experiments, we have used an implementation of Agnes (Agglomerative Nesting) that is fully described [14]. Agnes is run with single linkage (or single-link), meaning that we merge in each step the two clusters with the smallest minimum pairwise distance (which comes down to the nearest neighbor method).

One of the main weaknesses of the hierarchical clustering is that it requires a predefined number of output clusters. This is a problem for the WePS task as we do not know the number of different "real" persons that are covered by a person name in advance.

5. EVALUATION

For the evaluation of both clustering algorithms, we have compared our output against the gold standard that has been provided within the Web People Search competition framework. We have also used the evaluation metrics that are proposed for the WePS competition, being BCubed F-score (harmonic mean of BCubed Precision and Recall) and the Purity-Inverse Purity F-score (harmonic mean of Purity and Inverse Purity). Purity, as well as BCubed precision, refers to the frequency of the most common category in each cluster, and gives higher scores to clusters that introduce less noise. Inverse Purity, as well as BCubed recall, focuses on the cluster with maximal recall for each category, and gives higher scores for clusterings that group more elements of each category in a corresponding single cluster. For a detailed description of the evaluation metrics, we refer to Amigó et al. [1].

5.1 Data Sets

For the first WePS competition, the task organizers [2] provided the participants with trial, training and test data[7]. For the training set, the trial set was expanded in order to cover different degrees of ambiguity (very common names, uncommon names and celebrity names which tend to monopolize search results). The names were selected from the US Census corpus (32 names), from Wikipedia (7 names) and from the Program Committee listing of the ECDL-2006 conference (10 names). The Wikipedia and ECDL sets contain documents corresponding to the first 100 results for a person name query to the Yahoo! search engine[8], whereas the US Census sets contain a varying number of search engine results (from 2 to 405 documents) per person name. These documents were manually clustered and documents that could not be clustered properly were put in a "discarded" section. The test data were constructed in a similar way (30 sets of 100 web pages). Unfortunately, there was a general increase in ambiguity compared to the training set. The global ambiguity average (number of different entities per person name) is 10.76 for the training data, whereas for the test data it is 45.93 [2]. Given the largely different distributions in the training and test sets, this makes the task very challenging for a machine learning approach (e.g for training the distance threshold for clustering). For the second Web People Search competition [3], a new testbed of thirty

[7]Available at http://nlp.uned.es/weps

[8]All queries were performed with the Yahoo! API from http://developer.yahoo.com/search/web/

additional person names, improved evaluation metrics, and an additional attribute extraction subtask were created. We have trained and optimized our system on the WePS 1 training set, and used both test sets (from the two competitions) for the evaluation of both clustering algorithms.

5.2 Results on training data

Figure 1 shows the BCubed F-scores and Purity-Inverse Purity F-scores measured on the training data for both the Fuzzy Ants and hierarchical Agnes clustering on three different clustering thresholds (Agnes1: threshold 0.1, Agnes2: threshold 0.2 and Agnes3: threshold 0.3).

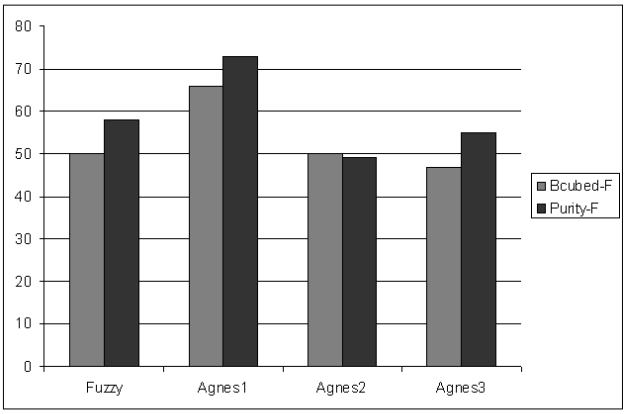

Figure 1: Training results for Fuzzy and hierarchical clustering (Agnes) on three different clustering thresholds.

The figures show a big impact of the clustering threshold for the Agnes algorithm, with F-scores that show differences of up to 20% (0.66/0.73 for the first threshold, 0.50/0.49 for the second threshold and 0.47/0.55 for the third threshold). The remarkable impact of the threshold on the Agnes clustering is important, as the threshold has to be predefined for each new data set, and for the Web people search task we can not specify the number of output clusters in advance.

The Fuzzy Ants clustering outperforms Agnes2 and Agnes3, but is beaten by the best Agnes clustering. Shallow error analysis shows that the algorithm has problems with data sets that have to be clustered in a few big clusters (e.g. all documents for Alan Hanbury have to be clustered in two large clusters). In addition, we notice very low average similarities between the documents that should be clustered together in this data set (average mean similarity of 0.08). Therefore it would be interesting to apply a rescaling process on the similarities, in order to obtain a more equal distribution of the similarity values, especially because the Fuzzy Ants algorithm performs better on equally distributed similarity values.

5.3 Results on test data

Figure 2 shows results for both the Fuzzy Ants and hierarchical Agnes clustering on three different clustering thresholds for the WePS1 testbed.

The Agnes results are less straightforward than for the

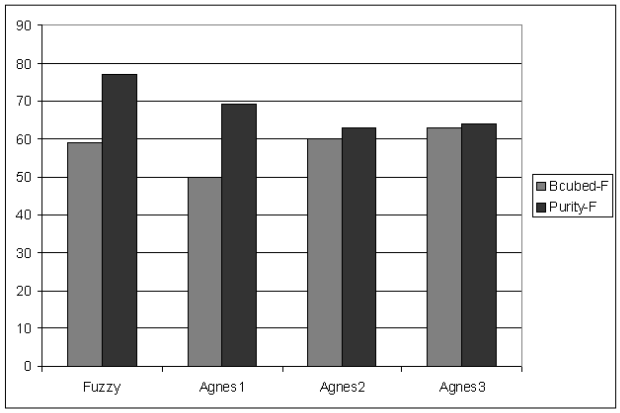

Figure 2: Results for Fuzzy and hierarchical clustering (Agnes) on three different clustering thresholds for the first WePS testbed.

training data: figures for threshold 1 outperform the other two for the Purity-Inverse Purity F-score, but are much worse than the others if we consider the BCubed F-score.

The Fuzzy Ants clustering, on the other hand, gives much better results than for the training data. This is probably due to the fact that this data set is much more balanced and ambiguous than the training set (See Section 5.1).

The results for both clustering approaches on the second WePS testbed are shown in Figure 3.

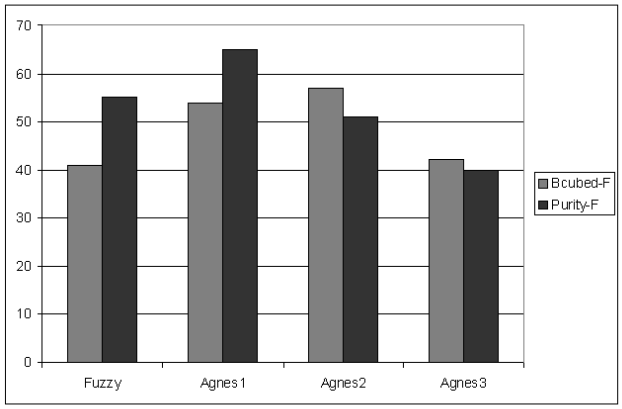

Figure 3: Results for Fuzzy and hierarchical clustering (Agnes) on three different clustering thresholds for the second WePS testbed.

This test set shows comparable tendencies to the training data: the first Agnes threshold outperforms the other two Agnes runs, as well as the Fuzzy Ants clustering. We encounter again very low average similarities (average mean similarity of 0.13 for documents that should be clustered together) and a couple of person names whose documents are clustered in one or very few big clusters. These persons obtain very low F-scores (David Tua: 0.02, Gideon Mann:

0.08) and have a big negative impact on the overall F-scores for this data set. A possible way of solving this problem would be a postprocessing step after the ants runs, where we calculate the similarities between singleton documents and the documents of larger clusters and compare these to the similarity between the centroid of the larger clusters and all documents of the large cluster. If a lot of the documents of the larger cluster seem very close to the singleton, we probably have to add the singleton to the larger cluster. The same process could be applied for merging smaller and larger clusters.

6. CONCLUSIONS

Since it is a priori not known how many persons are potentially referred to when performing a web people search task, the performance of most of the current clustering approaches suffers from their dependency upon a predefined number of clusters. In order to overcome this, we have experimented with a Fuzzy Ants clustering algorithm that has shown to be competitive with a classical hierarchical clustering algorithm on the second WePS testbed. It has the additional advantage that the number of clusters does not have to be decided or estimated a priori. Furthermore we believe there is room for improvement by rescaling the similarity values and adding a post-processing step on the Fuzzy Ants output.

7. REFERENCES

[1] E. Amigó, J. Gonzalo, and J. Artiles. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval Journal*, 2008.

[2] J. Artiles, J. Gonzalo, and S. Sikine. The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In *Proceedings of Semeval 2007, Association for Computational Linguistics*, 2007.

[3] J. Artiles, J. Gonzalo, and S. Sikine. Weps 2 evaluation campaign: overview of the web people search clustering task. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, April*, 2009.

[4] A. Bagga and B. Baldwin. Entity-based cross-document co-referencing using the vector space model. In *Proceedings of the 17th international conference on Computational linguistics*, pages 75–85, 1998.

[5] K. Balog, L. Azzopardi, and M. de Rijke. Personal name resolution of web people search. In *WWW2008 Workshop: NLP Challenges in the Information Explosion Era (NLPIX 2008)*, 2008.

[6] D. Bollegala, Y. Matsuo, and M. Ishizuka. Disambiguating personal names on the web using automatically extracted key phrases. In *Proc. ECAI 2006*, pages 553–557. Trento, Italy, 2006.

[7] W. W. Cohen. Fast effective rule induction. In A. Prieditis and S. Russell, editors, *Proceedings of the 12th International Conference on Machine Learning*, pages 115–123, Tahoe City, CA, 1995. Morgan Kaufmann.

[8] W. Daelemans and A. van den Bosch. *Memory-Based Language Processing*. Cambridge University Press, 2005.

[9] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. Mbt: A memory-based part of speech tagger generator. In *Proceedings of the 4th ACL/SIGDAT Workshop on Very Large Corpora*, pages 14–27, 1996.

[10] J. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain and, and L. Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behaviour*, pages 356–363, 1990.

[11] G. Gooi and J. Allan. Cross-document coreference on a large scale corpus. In *Proceedings of 42nd Annual Meeting of the Association for Computational Linguistics (ACL), Reference Resolution Workshop*, 2004.

[12] C. Hemphill, J. Godfrey, and G. Doddington. The atis spoken language system pilot corpus. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 96–101, 1990.

[13] D. Kalashnikov, R. Nuray-Turan, and S. Mehrotra. Towards breaking the quality curse. a web-querying approach to web people search. In *Proc. of Annual International ACM SIGIR Conference*, Singapore, July 20–24 2008.

[14] L. Kaufman and P. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, New York, NY, 1990.

[15] H. Kucera and W. Francis. *Computational analysis of present-day English*. Brown University Press, RI, 1967.

[16] E. Lefever, T. Fayruzov, and V. Hoste. A combined classification and clustering approach for web people disambiguation. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 105–108, 2007.

[17] E. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. In *From Animals to Animats: Proceedings of the Third International Conference on Simulation of Adaptive Behaviour*, pages 501–208, 1994.

[18] G. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of CoNLL-2003*, pages 33–40. Edmonton, Canada, 2003.

[19] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[20] Q. Ming Vu, A. Takasu, and J. Adachi. Improving the performance of personal name disambiguation using web directories. *Information Processing Management*, 2007. doi:10.1016/j.ipm.2007.11.001.

[21] N. Monmarché. *Algorithmes de Fourmis Artificielles: Applications à la Classification et à l'Optimisation*. PhD thesis, Université François Rabelais, Tours, France, 2000.

[22] T. Pedersen, P. A., and A. Kulkarni. Name discrimination by clustering similar contexts. In *Proceedings of the World Wide Web Conference (WWW)*, 2006.

[23] X. Phan, L. Nguyen, and S. Horiguchi. Personal name resolution crossover documents by a semantics-based approach. In *IEICE Transactions on Information and*

Systems, volume E89-D(2), pages 825–836. 2006.

[24] J. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[25] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th annual meeting on association for computational linguistics (ACL)*, pages 41–47. Morristown, NJ, USA, 2001.

[26] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. In *Information Processing Management*, volume 24(5), pages 513–523. 1988.

[27] S. Schockaert, M. De Cock, C. Cornelis, and E. Kerre. Clustering web search results using fuzzy ants. In *International Journal of Intelligent Systems*, volume 22, pages 455–474. 2007.