# Towards a Fair Comparison between Name Disambiguation Approaches

João Guerreiro, Daniel Gonçalves, David Martins de Matos
Technical University of Lisbon / INESC-ID Lisboa
Rua Alves Redol, 9. 1000-029, Lisboa, Portugal
joao.p.guerreiro@ist.utl.pt, {daniel.goncalves, david.matos}@inesc-id.pt

## ABSTRACT
Searching for information about people in search engines is a common and straightforward task that is often hampered by name ambiguities. While users are interested in information about a single person, results pages usually comprise many persons with the same name. There are several approaches to tackle personal name disambiguation; however, it is still a challenge to understand the impact of each approach alone. In this paper, we present a plugin-based framework that aims to compare and to identify the most promising approaches for name disambiguation. This framework enabled us to merge different approaches to find good combinations for this task and to compare state-of-the-art solutions using a common dataset. Preliminary results support the greater impact of biographical information to aid in clustering, the use of comprehensive texts instead of only metadata and TF-IDF instead of more complex approaches**.**

## Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information search and retrieval – *clustering, retrieval models, selection process*.

## General Terms
Algorithms, Measurement, Performance, Experimentation.

## Keywords
Personal Name Disambiguation; Vector Space Model; Plugin-Based Framework; Feature Selection; Clustering.

## 1. INTRODUCTION
Searching for information about people in search engines is a common and straightforward task. However, it is often troublesome due to the ambiguities that arise from name variations (e.g. "*John Smith*" and "*J. Smith*") and different people having identical names. When searching for common names, a results page contains information about different people, while users are interested in finding information about a single one.

Several researchers focused on name disambiguation in order to cluster the results by person. In these solutions, each group contains all the web pages about a different person. Most approaches to resolve name disambiguation rely on the Vector Space Model (VSM) to determine the similarity between documents. The use of TF-IDF with the cosine similarity measure and Hierarchical Agglomerative Clustering (HAC) is a very common solution. Yet, there are several variations in the features used, similarity measure, weighting or clustering method. The use of different datasets makes it harder to identify the best solution to tackle the name disambiguation problem. Apart from competitions such as the *WePs* workshops [1], the comparisons between techniques are usually limited to baseline methods.

In this paper, we present a plugin-based framework that aims at identifying the most promising approaches for name disambiguation. This framework has two main objectives. First, we want to merge different approaches in order to find the most promising solutions. Second, we want to compare state-of-the-art solutions using a shared dataset and making use of the same tools to back-up the experiments. This standardization enables us to specify which methods are responsible for differences in precision or recall. Moreover, the framework allows an easy introduction of new methods/tools to extract information from web pages; select the features; and similarity methods to cluster those web pages. We evaluated several techniques with a common dataset from the *WePS-2* clustering competition [1], which allowed us to also compare our results with theirs. In fact, several researchers have been making use of this dataset recently. Results support the use of the entire text in opposition to snippets, TF-IDF instead of more complex approaches and biographical data to enrich the weighting phase.

## 2. RELATED WORK
As noted earlier, the majority of name disambiguation approaches rely on the VSM, but several differences exist in all the phases it comprises. The first variation regards the data used from each document/webpage. Researchers resorted to the entire text, metadata, summaries or even specific sections. To cite one example, Bagga and Baldwin [2] produce summaries by extracting the text surrounding the person's name. Other differences arise from the features selected to represent each document. One may use all the words from the document, URLs or resort to part-of-speech taggers to find noun phrases or proper nouns. The use of named entities is very popular and it was reported to be one of the most efficient by several authors (e.g. [8]). A different, but very promising approach is to leverage biographical information such as birth year, occupation, birth location and e-mail (e.g. [5, 10]).

The use of TF-IDF (often with stemming) and the cosine similarity measure is very common in such systems. CU-COMSEM [4] presented a method that combines TF-IDF and Jaro-Winkler distance function, called Soft TF-IDF [6]. This


*OAIR'13*, May 22–24, 2013, Lisbon, Portugal.

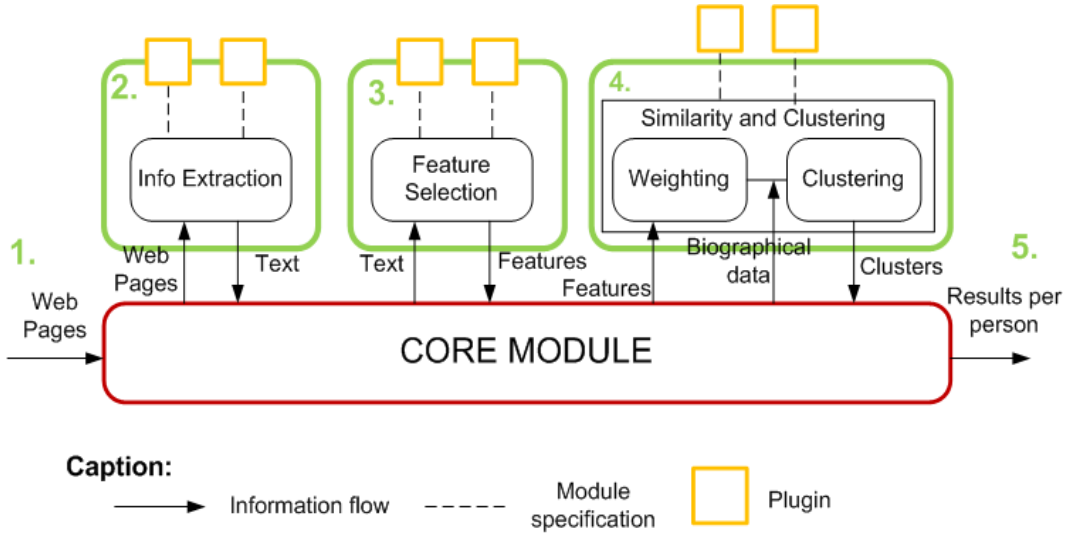


**Figure 1 – Structure and different steps of our plugin-based framework**

method aims to classify similar phrases as the same phrase. While stemming gets the root of words, this method also contemplates differences that arise from typing errors. However, it is much slower than TF-IDF. Other approaches match documents in the semantic space by making use of the relationships between terms. They assume that terms used in the same contexts probably have similar meanings (or are somehow related). Pedersen et al. [12] applied Latent Semantic Analysis (LSA) with bi-grams. LSA maps high-dimensional data to a lower dimensional representation in a latent semantic space, using Singular Value Decomposition (SVD) [9]. Song et al [15] investigated the Probabilistic LSA (PLSA) [7], which offers further statistical foundation to the LSA method. It is based on the probability of documents being related with latent topics (documents generate a particular distribution of topics), and those topics being related with words (documents generate a particular distributions of words). Song et al also explored Latent Dirichlet Allocation (LDA) [3], a three level Bayesian Hierarchical method. In LDA, the topic distribution is assumed to have a Dirichlet Prior that remains the same for all documents in a collection (the latent variables are not dependent on each document). Although semantic techniques are promising, they have been unable to outperform non-semantic approaches.

There are many approaches and systems trying to disambiguate names; yet, it is difficult to compare them fairly and merge them with little effort. At an early stage of our research, we are exploring methods based on the VSM using HAC, but there are several approaches used to disambiguate names (e.g. [11, 16]).

# 3. PLUGIN-BASED FRAMEWORK AND IMPLEMENTED TECHNIQUES

We have implemented a Python[1] plugin-based framework for name disambiguation that allows an easy comparison between promising methodologies and combinations within. It is based in modules encompassing plugins (Figure 1). With this architecture, we may insert a new methodology without modifying any existent code. This is valid for the web page information extraction (in 2), features selection (in 3) and weighting and clustering (in 4). The core module works as a connection between the different phases.

We aim to compare the most promising approaches and identify the effect of each one of them alone. To improve clarity with an example, we may maintain the whole setting and vary only TF-IDF and LSA. This will indicate the gain/loss that these methods provide without external influence. In the next sections, we will describe each module and specify the implemented techniques.

## 3.1 Information Extraction

This module extracts the information from each webpage. Each plugin may deal with this issue differently. We may vary the method (e.g. all text or specific section) or experiment different tools with the same purpose. Herein, we implemented *Metadata Extraction* and *Body Text Extraction*. The former was provided by the *WePS-2* clustering task and included the titles and snippets of each result. The latter extracted the text in the body of each web page using the tools *Beautiful Soup*[2] and *BodyTextExtractor*[3].

## 3.2 Feature Selection

The feature selection module intends to find a rich set of features that represent each document in the VSM. Each feature type may be used alone or in combination with others. The biographical features are not included in the VSM. Instead, they are used in the following model since they have bigger influence on the clustering task. Based on the related work, we resorted to the following feature types: *all words/tokens*, *terms*, *named entities*, *only nouns* and *biographical features*.

**All Words**. To tokenize our documents we used the *topia.termextract*[4] module. The stop words were removed and we provided an option to stem the tokens.

**Terms**. We used the *topia.termextract* module. It uses a POS Tagger and makes use of simple statistics and linguistics to produce lists of terms (wider list than *Named Entities*).

**Named Entities**. They were mentioned as rich features. We used the *Illinois Named Entity Tagger* [13], since it presented, to our knowledge, the best performance for publicly available *Named*

[1] http://www.python.org/ (Last visited: 03/2013)

[2] http://pypi.python.org/pypi/BeautifulSoup (Last visited: 03/2013)

[3] http://github.com/aidanf/BTE (Last visited: : 03/2013)

[4] http://pypi.python.org/pypi/topia.termextract/ (Last visited: 03/2013)

*Entity Recognizers* (NER). It tags the text with 4 types of named entities (people, organizations, locations and miscellaneous) and has a large set of training data.

**Nouns**. We tried three POS Taggers from *NLTK*[5], *topia.termextract* and *TreeTagger* [14]. We selected *TreeTagger*, because it provided the best results in pilot experiments.

**Biographical Information**. Some evidence support that biographical attributes can improve personal name disambiguation [5, 10]. The attributes we used were the person's occupation, related organizations, birth year and name (names with more information than the query). These attributes gain more preponderance because they can provide more confidence to a match. For example, if two webpages mention the same birth year and occupation, it is very probable that they are the same person, so the similarity value highly increased. In contrast, the value may decrease if attributes such as name and birth year differ. To extract those attributes from the text we used hand-coded patterns for birth year and name, the *NER* tool for the organization and a dictionary of occupations.

**Table 1 – Some results from information extraction**

| | Approach | Precision | Recall | F0,5 |
|---|---|---|---|---|
| LSA - all tokens | Metadata | 0.79 | 0.47 | 0.52 |
| | Body text | 0.81 | 0.49 | 0.54 |
| TF-IDF - all tokens and biographic | Metadata | 0.89 | 0.67 | 0.74 |
| | Body text | 0.77 | 0.70 | 0.71 |
| TF-IDF – Named Entities | Metadata | 0.99 | 0.43 | 0.54 |
| | Body text | 0.98 | 0.48 | 0.58 |

## 3.3 Similarity and Clustering

This module weights the matrix feature-document, measures the similarity (with the cosine distance measure) and performs the clustering. To weight the matrix feature-document and calculate similarities we used *Gensim*[6], a *Python Framework* that supports VSM. This framework supported the experiments with TF-IDF, LSA and LDA. The HAC was performed with the *hcluster module*[7]. The clustering method starts from singleton clusters for each webpage and merges sequentially the closest clusters according to a certain threshold. We used single-link, complete-link and group-average methods to cluster. Single-link calculates the distance via the two closest members of each cluster; complete-link considers the most distant values; and the group-average considers all the elements calculating an average.

# 4. EVALUATION AND RESULTS

We evaluated the aforementioned techniques to understand the impact of each one of them. We run several combinations among approaches using a common dataset. We used the corpus from the *WePS* 2009 competition [1], which comprises the top 150 search results for 30 different names. The objective was to cluster the web pages by person and compare the results with a manually annotated gold standard. We applied the frequently used B-Cubed scoring method [2] and calculated the F-measure (based on precision and recall). In what follows, we present comparisons between techniques and highlight the most promising ones.

## 4.1 Metadata Vs Body Text

We performed several experiments where we varied these two extraction methods. Most of the results were slightly worse when using metadata (a few examples in Table 1). It was interesting to notice that poorer sets of features (e.g. all tokens) provided similar results (e.g. LSA – all tokens), but the differences increase in favor of *Body Text* when using *named entities* or *terms*. It is explained by the absence of many of these features in the snippets. Although the *Body Text* provides better results in general, the best result occurs when using the *Metadata* with TF-IDF, all tokens and biographical information. The decrease of precision for the *Body Text* condition indicates that some biographical attributes were incorrectly assigned, probably fitting in other person mentioned in the webpage. In contrast, the *Metadata* has a succinct description more related with the person's name.

**Table 2 – Some results from feature selection**

| | Approach | Precision | Recall | F0,5 |
|---|---|---|---|---|
| **Body text and TF-IDF** | All tokens | 0.95 | 0.48 | 0.57 |
| | Only NEs | 0.98 | 0.48 | 0.58 |
| | NEs + Biograph. | 0.79 | 0.71 | 0.72 |
| | Only terms | 0.99 | 0.43 | 0.53 |
| | All tokens Biograph. | 0.77 | 0.70 | 0.71 |
| | NEs+ Nouns | 0.97 | 0.47 | 0.56 |
| | Only Nouns | 0.95 | 0.46 | 0.55 |
| | Terms + NEs | 0.99 | 0.45 | 0.56 |

## 4.2 Feature Selection

We evaluated each feature alone and a few combinations. Table 2 shows the results when using the *body text* and TF-IDF, but other approaches presented similar conclusions. An exception occurs for features such as named entities or terms, that decrease their performance when using the metadata instead of the body text due to the lack of features in that text. The use of biographical information clearly enhanced the performance, due to a boost on recall (more than 20 percentage points - pp). Yet, precision was also affected by this method. It shows that we extracted incorrect information, probably related to other people mentioned in the webpage. Yet, we assume that the methods to extract this information can be highly improved. Despite this feature, named entities presented the best results; still, the difference is small.

**Table 3 – Comparison - TF-IDF Vs semantic approaches**

| | Approach | Precision | Recall | F0,5 |
|---|---|---|---|---|
| **Body text – All tokens** | TFxIDF | 0.95 | 0.48 | 0.57 |
| | LSA | 0.81 | 0.49 | 0.54 |
| | LDA | 0.82 | 0.50 | 0.55 |
| **Body Text – Only Named Entities** | TFxIDF | 0.98 | 0.44 | 0.58 |
| | LSA | 0.88 | 0.43 | 0.52 |
| | LDA | 0.86 | 0.45 | 0.52 |
| **Body Text – Named Entities + Biographic** | TFxIDF | 0.79 | 0.71 | 0.72 |
| | LSA | 0.78 | 0.71 | 0.72 |
| | LDA | 0.70 | 0.74 | 0.69 |

## 4.3 TF-IDF Variations and Semantic Options

The comparison among TF-IDF variations did not point towards the use of Soft TF-IDF. The regular TF-IDF presented the worst

[5] http://www.nltk.org/ (Last visited: : 03/2013)

[6] http://nlp.fi.muni.cz/projekty/gensim/ (Last visited: 03/2013)

[7] http://pypi.python.org/pypi/hcluster/ (Last visited: 03/2013)

results, but very close to Soft TF-IDF (1 to 2pp - $F_{0.5}$). TF-IDF with Stemming performed better (gained from 3 to 5pp - $F_{0.5}$). Indeed, we used this TF-IDF variation in the other comparisons. Semantic approaches (LSA and LDA) seemed promising approaches for this context, but were short of expectations in related work. In fact, it is supported by a comparison with TF-IDF (Table 3). LSA and LDA had very similar results between them, but were always slightly worst than TD-IDF, which seems to be the best choice; it holds the best results and is faster.

## 4.4 Clustering Options

In the previous comparisons, we used the *group-average* link to merge the clusters; however, we performed several experiments where we varied the threshold and linkage option to merge the clusters. Overall, *single-link* provided the best results, as it heightens the recall. Yet, further experiments have to be done due to the impact of the threshold used.

## 4.5 Comparison with WePS-2 Systems

The use of the *WePS-2* dataset enabled a comparison between our approaches and the systems that participated in the competition (Table 4). The victor system (*PolyUHK*) supports our results due to their focus on biographical information. Besides the boost supported by the recall, they were able to maintain high precision values. It also explores the use of bigrams and the detection of the type of page (e.g. in a homepage, it considers the text near the word "I"). The second place used a simple but effective approach, using single-link clustering (confirmed by our evaluation) and a normalized TF-IDF. The third place used two-stage clustering; which is known to provide good results. Our best result achieved the fourth position; yet, the extraction of biographical information can be highly improved to provide better results.

**Table 4 – Comparison with WePS-2 systems**

| Rank | System | Precision | Recall | F0,5 |
|---|---|---|---|---|
| 1 | PolyUHK | 0.87 | 0.79 | 0.85 |
| 2 | UVA_1 | 0.85 | 0.80 | 0.81 |
| 3 | ITC-UT_1 | 0.93 | 0.73 | 0.81 |
| - | **OUR BEST** | 0.89 | 0.67 | 0.74 |
| 4 | XMEDIA_3 | 0.82 | 0.66 | 0.72 |

# 5. CONCLUSIONS

Name disambiguation is still an open research area. There are several techniques and tools to tackle this problem, but it is very difficult to compare them fairly. Our plugin-based framework allowed us to implement several state-of-the-art approaches, to merge them in order to find the most promising solutions and to compare them. This comparison enabled us to identify the impact of each method in separate. In a nutshell, using the entire text was better than using metadata (snippet and title); the use of biographical information highly increases the performance; and TF-IDF outperformed more complex approaches. Our best result was the combination between two techniques that were popular in this experiment (TF-IDF and biographical data) and other not so popular ones (*Metadata* and *All Tokens*). This comparison provides an important contribution to the name disambiguation problem. Moreover, this framework paves the way to more extensive comparisons, which may include other techniques such as two-stage clustering and the use of bigrams. In addition, it will allow us to address the impact of other factors such as the threshold used in clustering.

# 6. ACKNOWLEDGMENTS

This work was supported by the Portuguese Foundation for Science and Technology (FCT): individual grant SFRH/ BD/66550/2009; project PAELife AAL/0014/2009; and project PEst-OE/EEI/LA0021/2011.

# 7. REFERENCES

[1] Artiles, J., Gonzalo, J., Sekine, S. (2009). Weps 2 evaluation campaign: overview of the web people search clustering task. In WePS 2 Evaluation Workshop. WWW Conference 2009.

[2] Bagga, A. and Baldwin, B. (1998). Entity-based cross-document coreferencing using the vector space model. Proc. of the international conference on Computational linguistics.

[3] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. J. Mach. Learn. Res., 3:993–1022.

[4] Chen, Y. and Martin, J. (2007). Cu-comsem: exploring rich features for unsupervised web personal name disambiguation. In proc. of SemEval '07.

[5] Chen, Y, Lee, SYM, and Huang, C.-R. (2009). Polyuhk: A robust information extraction system for web personal names. In 2nd WePS, 18th WWW Conference.

[6] Cohen, W., Ravikumar, P., and S., F. (2003). A comparison of string metrics for matching names and records. Proc. of the IJCAI Workshop on Information Integration on the Web.

[7] Hofmann, T. (1999). Probabilistic latent semantic indexing. In SIGIR '99: Proceedings of the ACM SIGIR conference on Research and development in information retrieval.

[8] Ikeda, M., Ono, S., Sato, I., Yoshida, M., and Nagawaka, H. (2009). Person name disambiguation on the web by twostage clustering. In 2nd WePS, 18th WWW Conference.

[9] Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. In Discourse processes, pages 259–284.

[10] Mann, G. S. and Yarowsky, D. (2003). Unsupervised personal name disambiguation. In Proc. of conference on Natural language learning at HLT-NAACL, NJ, USA. ACL.

[11] Nuray-Turan, R., Kalashnikov, D. V., and Mehrotra, S. (2012). Exploiting web querying for web people search.*ACM Trans. Datab. Syst*. 37, 1, 41 pages.

[12] Pedersen, T., Purandare, A., and Kulkarni, A. (2005). Name discrimination by clustering similar contexts. In International Conference on Intelligent Text Processing and Computational Linguistics, pages 220–231. Springer.

[13] Ratinov, L. and Roth, D. (2009). Design Challenges and Misconceptions in Named Entity Recognition. In CoNLL.

[14] Schmid, H.(1994). Probabilistic part-of-speech tagging using decision trees. In Proc. of the International Conference on New Methods in Language Processing, 1994.

[15] Song, Y., Huang, J., Councill, I., Li, J., and Giles, C. (2007). Efficient topic-based unsupervised name disambiguation. Proc. of ACM/IEEE-CS joint conference on Digital libraries.

[16] Tang, J. et al. (2012) A unified probabilistic framework for name disambiguation in digital library. *Knowledge and Data Engineering, IEEE Transactions* on 24.6: 975-987.