Novel Techniques for Text Annotation with Wikipedia Entities

Christos Makris, Michael Simos

Novel Techniques for Text Annotation with Wikipedia Entities

Christos Makris, Michael Angelos Simos

Computer Engineering and Informatics Department,
University of Patras, Greece

{makri,asimos}@ceid.upatras.gr

Abstract. Text annotation is the procedure of identifying the semantically dominant words of a text segment and attaching them with conceptual content information in their context. In this paper, we propose novel methods for automatic annotation of text fragments with entities of Wikipedia, the largest knowledge base online, a process commonly known as *Wikification* aiming at resolving the semantics of synonymous and polysemous terms accurately. The cornerstone of our contribution is a novel iterative Wikification approach, converging at optimal annotations while balancing high accuracy with performance. Our first two methods can be fine-tuned through a machine-learning technique over large homogenous data sets. Our experimental evaluation resulted in remarkable improvement over state-of-the-art Wikification approaches.

Keywords: Text semantic annotation, Wikipedia entities, Semantic Data Linking on Web, Data Mining, Information Retrieval, Ontologies

1 Introduction

Wikipedia is one of the largest online knowledge repositories, consisting of more than 4 million entities (also called lemmas or articles), built over more than with over 20 million contributing users. The online "crowdsourcing" nature of Wikipedia enables the creation and maintenance of knowledge entities in a quite diverse manner resulting in widespread and commonly accepted textual descriptions, through the consensus of a large number of people. Wikipedia inherits the main principles of web, elegantly combining textual content and cross-references with hyperlinks providing plenteous semantic information that can be exploited for representing the described entities.

Text annotation is the process of identifying the semantically dominant words (called spots or anchors) in a text segment and attaching them with additional information expressing conceptual content in their current context. Recent research works [1, 3, 4, 5, 6, 9, 12, 13, 15] explored various approaches for efficient text annotation with Wikipedia entities. Text annotation is a fundamental preprocessing step of many information retrieval activities like semantic indexing, cross-referencing content on web, clustering, classification, for efficient rankings, text summarization, natural lan-

guage generation, exploiting conceptual similarity of documents, and sentiment analysis.

Text annotation techniques face the entity disambiguation problem, where multiple candidate Wikipedia article annotations exist for a specific term, deriving by polysemy and synonymy, common properties of natural languages. Disambiguating to Wikipedia entities is similar with classic Word Sense Disambiguation tasks [14], but also requires compatibility with additional information provided by the link structure of Wikipedia. The core of most existing methods involves the textual context and the collective agreement of other identified spots with every candidate disambiguation of a specific spot.

The aim of current work is the introduction of simple and accurate disambiguation methods for small texts, using Wikipedia as the underlying catalogue, thus focusing on the same kind of texts as in TAGME system [3] and the approaches introduced in [11, 13].

2 Literature Overview

Wikipedia disambiguation techniques can be distinguished into two main classes: local and global [15]. Local approaches exploit context information concerning only the specific spot, while global approaches rely on a collective agreement of all spots in a text segment and their disambiguation. Moreover, some approaches involve collection-level methods, performing a collective disambiguation of all entity references across a given document collection.

Disambiguation based on Wikipedia entities has attracted many researchers [1, 9, 11, 12, 13, 15], with most prominent the approach proposed in [3], where the authors focused on real time annotation of short texts existing on tweets, web search engine results etc, where textual context is quite limited. The TAGME system was extended in [10] by using information from knowledge resources such as WordNet [2], and exploiting the PageRank values inherent in the Wikipedia pages. The system can be extended to handle other ontologies [7] as well.

3 Proposed Methods

The touchstone of our proposed techniques is TAGME's voting scheme that assigns to each candidate sense of a spot a value, based on the collective agreement of this annotation from the other spots in the same text. Our first approach investigates some minor changes over TAGME's scheme. The second method proposes a more analytical model for the problem. Finally our iterative method, introduces a novel approach.

Spots Extraction

We use a simple yet very effective method for preprocessing the input text fragments for the extraction of candidate Wikipedia entities. Firstly, every text fragment

is separated into tokens (words, punctuation, symbols, etc) to be grouped later, forming phrases when possible. The grouping is performed by matching as many adjacent tokens as possible with phrases (token groups) extracted from Wikipedia's links anchors. Complex rules have been applied allowing loose matching, handling minor details and further improving our results. The final step of spots extraction process is the removal of words not containing semantic information, (stopwords), symbols, punctuation etc. A more complex morphosyntactic analysis model is to be developed as future work.

3.1 Method 1

Let a_i be an anchor to be annotated, and $Pg(a_i)$ the list of candidate Wikipedia articles for this annotation, derived from occurrences of a_i in Wikipedia hyperlinks as anchor to the above articles. A vote provided by a spot c to the annotation $a_i \rightarrow p_j$ with p_j in $Pg(a_i)$ is evaluated as:

$$vote_c(p_j) = \sum_{p_l \in p_g(c)} srel(p_j, p_l) * P(p_l|c) \;/\; |Pg(c)|$$

where $P(p_l|c)$ is the prior probability of c is pointing to p_l (also known as commonness). Commonness is pre-calculated by parsing the Wikipedia dataset and for each observed anchor-Wikipedia pair dividing the number of appearances that the anchor points to the corresponding article by the total appearances of the anchor. The quantity $srel(p_j, p_l)$ is a measure devised by us to depict the relatedness between two Wikipedia pages that is computed by taking into account the commonality of the incoming Wikipedia links to the two pages, as the cardinality of intersection divided by the maximum cardinality set, thus:

$$srel(p_j, p_l) = \frac{|in(p_j) \cap in(p_l)|}{max\,(|in(p_j)|, |in(p_l)|)}$$

where in(p) as in [2, 4] denotes the set of Wikipedia pages pointing to page p.
The disambiguation weight of a probable annotation $a_i \rightarrow p_j$ is defined as:

$$score_{a_i}(p_j) = \sum_{a_t \neq a_i} vote_{a_t}(p_j)$$

The final annotation is selected based on maximum commonness score after a filtering step among Wikipedia articles within a voting score distance threshold τ from the maximum voting score.

We have tested relatedness formulas like [5, 13], but the highest accuracy is achieved on this schema by our simple formula (srel), evaluating the incoming link intersection of two articles. A common observation throughout our experiments is that the heterogeneity of compared articles in size, results in imbalances of relatedness score, when using approaches like [5] and [13]. A better formula is to be developed as future work.

3.2 Method 2

Let $Pg(a_i)$ be the set of candidate Wikipedia articles ($p_{a_i}j$) annotations for each spot as anchor a_i. The anchor-Wikipedia pairs that appear less than three times, or in

less than 0.1% of total occurrences of the anchor are filtered. Let a text fragment contain *k* spots a_i where $i \in \{1, 2, \ldots, k\}$.

The core logic of this method involves computing a global score for every probable combination of anchor senses represented as possible Wikipedia article annotations (p_{a_i}). There are $\prod_1^k \binom{|Pg(a_i)|}{1} = \prod_1^k |Pg(a_i)|$ probable combinations, that can be reduced by filtering more anchor-Wikipedia low probability pairs. Let *Gscore* be the global evaluation score of an annotation combination, as described above:

$$Gscore\left(p_{a_1}, p_{a_2} \ldots p_{a_k}\right) = \sum_{i=1}^{k-1} \sum_{w=i+1}^{k} Bscore(p_{a_i}, p_{a_w})$$

At first in "Method 2 (*rel*)" we evaluated a relatedness measure between two Wikipedia Articles:

$$rel(p_j, p_l) = 1 - \frac{log\left(max\left(|in(p_j)|, |in(p_l)|\right)\right) - log\left(|in(p_j) \cap in(p_l)|\right)}{log(W) - log(min(|in(p_j)|, |in(p_l)|))}$$

proposed by Milne & Witten [13] as the *Bscore* formula, where W denotes the size of Wikipedia. Still our simpler relatedness approach (*srel*) outperforms this commonly utilized relatedness measure yielding more accurate results, for most entities pairs having small incoming links intersection sets.

Finally, in "Method 2 comrel" we tested a more complex formula for *Bscore*:

$$Bscore\left(p_{a_i}, p_{a_w}\right) = comrel\left(p_{a_i}, p_{a_w}\right) = srel\left(p_{a_i}, p_{a_w}\right) P(p_{a_i}|a_i) P(p_{a_w}|a_w)$$

where $P(p_{ai}|a_i)$ is the Commonness of the anchor a_i pointing the article p_{ai}. This way *BScore* scales respectively with the commonality of the two articles combination.

As a final step, again we perform filtering by keeping all candidate combinations achieving a *Gscore* up a threshold distance τ from the maximum *Gscore*. Among the filtering results, we select the combination maximizing the total commonness $\sum_{i=0}^{\kappa} P(p_{a_i}, a_i)$. For very small threshold values (*t*), the selection of annotations yielding the highest Bscore is voted as the dominant. If the threshold is large, then a selection with the highest total commonness score is selected.

Threshold Optimization.

The most important fine tuning in our methods schemas involves the optimal selection of τ. We propose a method for optimizing τ threshold, given a Wikipedia entity manually annotated training set, having homogenous statistics with the to-be-annotated dataset. This technique can also be used to determine experimentally the upper bound of performance for our methods (by training with our experimental dataset).

Based on this optimization technique we explore the threshold value intervals for which a selection with the optimal annotations passes through the final filtering step. Then we just find the values interval for τ, that belongs to the maximum number of each selection interval, for the selections that perform a correct annotation.

Thus, the problem of optimizing τ value can be deducted to a simple optimization problem of selecting a value that belongs to as more intervals as possible, by simply checking the values on intervals edges.

3.3 A Novel approach on text annotation.

Various model schemas have been introduced for the text annotation problem in previous works with most common the voting scheme and its many variations, to which we contributed during the previous part of this paper. One of the most important breakthroughs of the current work is the introduction of a novel method for semantic anchor annotation with Wikipedia entities, through an iterative approach, aiming at converging to an optimal candidate for each anchor as a result of constant improvement of the approximate solutions of each iteration. The principal intuition of this method, lies on the analysis of human semantic interpretation process of text segments containing polysemous terms. Such a process would involve an iterative evaluation procedure, until some termination criteria are met. For the most of the common text segments disambiguation, those criteria may be met before iterations are required, but in complex polysemous context, an evaluation of each candidate annotation may be necessary, consisting an iterative procedure until some criteria for a decision with a degree of certainty are met. The main logic difference of this method and method 2, which evaluates every possible annotation combination among anchors of a text segment, is the targeted evaluation of semantically meaningful combinations (as evaluated by some commonness and relatedness formulas), resulting a vast complexity reduction.

A candidate model of such a process may be epitomized by the following iterative method:

Let $s_0, s_1, s_2, \ldots s_n$ be the spots of a text fragment, and $P_{si}1 \ldots P_{si}m \in Pg(s_i)$, the m candidate Wikipedia entity annotations of spot s_i, where $i \in \{0,1,2,3\ldots n\}$. Let $Pg(s_i)$ the set of candidate Wikipedia entity annotations of spot s_i. Each state is constituted of lists of candidate Wikipedia entity annotation of each spot sorted by a ranking criterion. At initial state, the candidate annotation lists are sorted by commonness in Wikipedia where $p_{si}0$, the most common entity of each spot s_i).

s_0	s_1	s_2	…	s_n
$p_{s0}0$	$p_{s1}0$	$p_{s2}0$		$p_{sn}0$
$p_{s0}1$	$p_{s1}1$	$p_{s2}1$		$p_{sn}1$
$p_{s0}2$	$p_{s1}2$	$p_{s2}2$		$p_{sn}2$
…	…	…	…	…
$p_{s0}m_0$	$p_{s1}m_1$	$p_{s2}m_2$		$p_{sn}m_n$

So the disambiguation approximation during the initial step of the iteration is $p_{s0}0$ $p_{s1}0$ $p_{s2}0$ … $p_{sn}0$, equivalent with the commonness annotation approach. The iterative step involves sorting each of the spots list in descending order of its elements R metric values. Every iteration results in approximation improvement of the optimal annotation entities, by exploiting as initial seed the commonness of the candidate entities,

and combining relatedness at each iteration step until convergence. The evaluated formula for the *R* metric is:

$$R(p_{s_i}x) = \sum_{w=0}^{n} srel(p_{a_i}, p_{a_w}) * P(p_{a_i}|a_i) * P(p_{a_w}|a_w)$$

$$Where: srel(p_i, p_l) = \frac{|in(p_j) \cap in(p_l)|}{max\left(\left|in\left(p_j\right)\right|, |in(p_l)|\right)}$$

The iterative algorithm is terminated when one of the following convergence criteria is met:

- A maximum number of iterations is reached. This is more of a security criterion, for handling the case of infinite iterations, due to other convergence criteria not being met. It is rarely activated, since in most cases, the rest of the convergence conditions are already met. The enforcement of large values results in more accurate results, relying more and more on relatedness for the cases where one of the other termination conditions is not yet satisfied. The application of smaller values, weights commonness more over relatedness, thus results in faster overall and worst case execution time, with an expense in accuracy.
- There are zero disturbances during the last *k* iterations. This means that a convergence is due to happen, since no changes on the list rankings are observed.
- The top m elements of each spots list maintain their position for *k* iterations. Thus the probability of disturbances is decreased.
- Small difference of each of the spot lists elements values is observed between iteration steps, stabilizing the ranking and leading to convergence.

The convergence speed of the above criteria can be enhanced by filtering the lower *k* percentile elements of each lists spots, after *m* iterative steps, since our evaluation revealed a fairly decreased probability of the lowest *k* elements occupying top positions on the lists rankings, before some of the convergence criteria is satisfied. This behavior is intuitively explained by the fact that for the vast majority of cases, the ranking results tend to converge at optimal values through each iteration.

The real advantages of the third method, involve remarkable improvements compared with our two first contributions. Its iterative nature, allows balancing between speed and accuracy, with the appropriate fine tuning of the convergence conditions parameters. Flaccid convergence criteria lead to more mature results in terms of relatedness, approximating at most cases the accuracy of our second method. Tighter convergence criteria imply improved average and worst case time complexity, without significant sacrifices in accuracy. Finally we should note that convergence occurs within the first iterations in most cases with high accuracy, a fact that allows us conclude correlation between convergence speed (in iterations) and the degree of certainty of the results accuracy, as is intuitively expected.

We exploited the above correlation among convergence speed and accuracy, by composing a post processing pruning step and evaluating the experimental results during our evaluation.

4 Experimental Evaluation

For the experimental evaluation of the methods we introduced above, we used the datasets of TAGME [3], available online by the authors. Our principal focus involved the Wiki-Disamb30 data file consisting of 1.4M short texts randomly selected from Wikipedia pages, each one containing approximately 30 words and at least one ambiguous anchor. Common precision metrics were used as in [13, 9, 2], in order to evaluate in practice the performance of our algorithms.

In table 1 and figure 1, we present the precision of methods 1 and 2, after optimization of their parameters after training. The optimal result column of table 1, presents the maximum possible accuracy of the method, using optimal parameters, calculated through training with the entire evaluated dataset. Figure 2, shows the precision of method 3, by varying the number of max iterations parameter. Figure 3 presents the evaluation of method 3 pruning step (precision/recall of pruning). The evaluation of method 3 is presented separately from methods 1 and 2, since it doesn't involve a machine learning procedure.

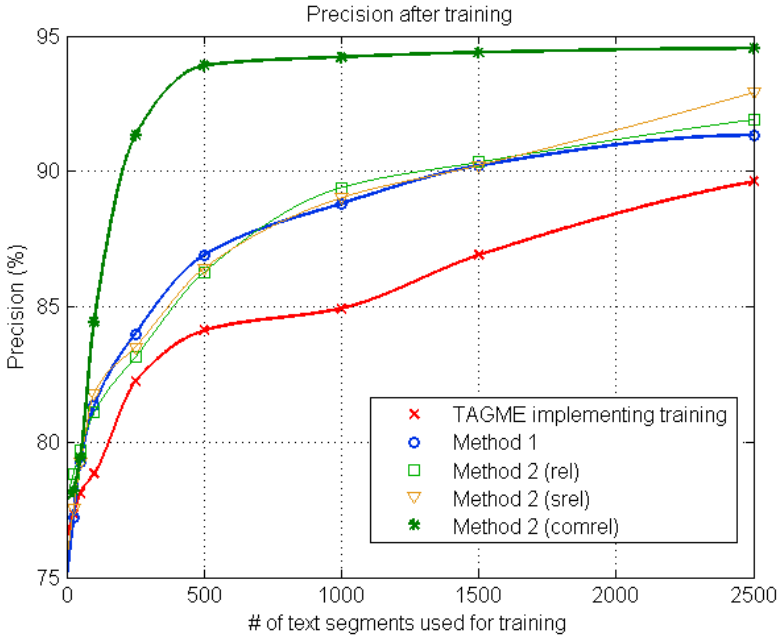

Figure 1

We used an Ubuntu Server 12.04LTS on a quad core 2.9 GHz 64 bit PC utilizing 8GBytes of main memory. The proposed methods were implemented in Python 2.7.6 over the Wikipedia dump files instance enwiki-20130503, loaded on a PostgreSQL database.

We ran our methods on randomly selected subsets of Disamb30 dataset, since the time complexity of method 2 was beyond our time resources for a full scale experiment, having in all cases similar results. The results, provided below are over a subset of more than 10K disambiguated spots. We also performed a simplified implementation of TAGME disambiguation algorithm, as described in [3] as performance baseline, applying a similar threshold optimization technique of its ϵ value.

Training	250 text fragments	Optimal
TAGME	77.53	91.31
Method 1 (rel)	81.28	92.43
Method 1 (srel)	81.95	92.92
Method 2 (rel)	80.19	92.07
Method 2 (srel)	80.59	92.98
Method 2 (comrel)	91.29	94.61

Table 1

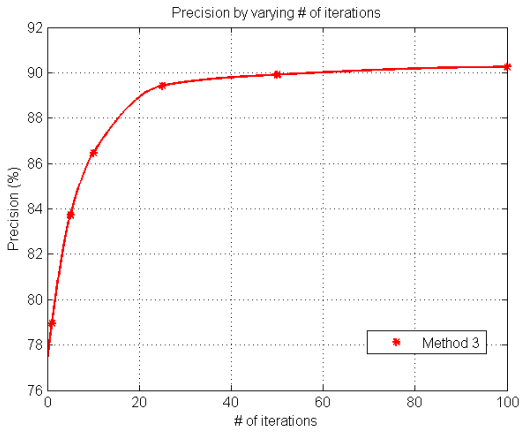

Figure 2

Table 1 and figure 1 present our detailed results on the precision of the two first methods proposed, expressing their capacity to give the same senses/Wikipedia articles initially attached to the text fragments anchors. Throughout our experiments, we investigated the upper bound of our methods precision while annotating every available anchor of the dataset.

Our first method outperforms TAGME, yielding more accurate optimal results up to 92.43%. This improvement is an outcome of the use of *srel* formula, depicting the necessity of a better relatedness measure between two Wikipedia entities for similar annotation algorithm schemas. The behavior of this method under our *τ-optimization* technique was also improved comparatively with TAGME, converging faster near optimal results.

The second method reached optimal accuracy of 94.61%, but the time complexity of this method was by far larger than both our first method and TAGME. The rel and srel formulas were not as effective, yielding respectively 92.07% and 92.98%, optimal accuracy. But when using *comrel* formula, the improvement in both accuracy and training set size for fast convergence to optimal results was remarkable. This fact was analyzed in depth, concluding that the analytic model of Method 2 creates a high ranking quality pre-filtering list, with the correct article of an annotation in the top of that list in most cases. That leads to very small optimal threshold values (τ) selection, thus fast convergence to the optimal τ. In conclusion, we note the necessity of weighting commonness with relatedness by multiplying as our *comrel* formula has proven.

Finally we evaluated our iterative method's precision, (figure 2), varying the maximum iterations parameter. The optimal results for this method's accuracy exceeded 90.71%. We have not developed an optimization technique for the current method due to its numerous parameters, thus its complexity. We applied common precision and recall metrics for the overall evaluation including the pruning step (figure 3), yielding exceptional conclusions for the quality of pruning.

To conclude, method 3 yields less accurate results from the previous methods optimal results, when annotating all available anchors in a text fragment. Yet the optimal accuracy of the current method, is to be explored in future but consist a complex task. In very low pruning recall conditions, accuracy exceeds 98%, with ambiguous anchors contributing as well in these high results. Method 3 accuracy outperforms limited training versions of methods 1, but still remains less accurate than method 2. Method 3 balances speed with accuracy, with the ability to adjust this equilibrium, yet with no expectancy of outperforming method 1 in speed, or method 2 in accuracy.

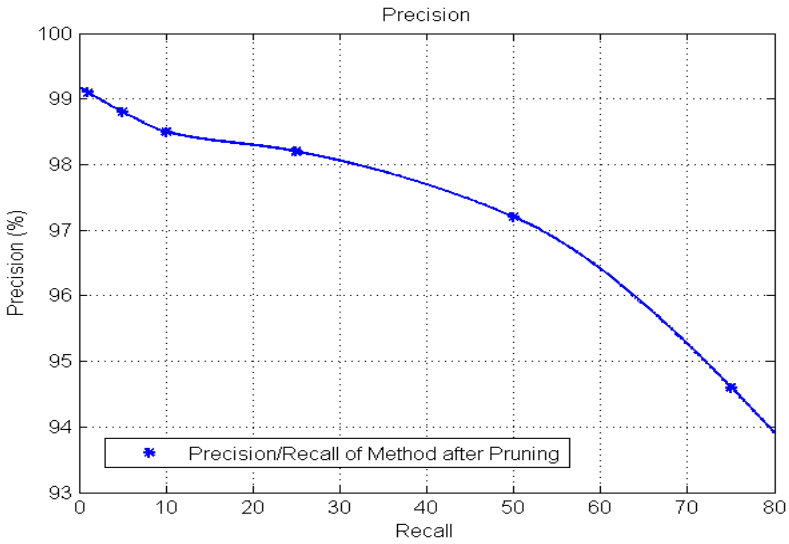

Figure 3

5 Conclusion

In this paper we propose three novel methods. The first method, revealed the potentiality for further improvement of the relatedness measures between Wikipedia entities. Our second method yields very accurate results, yet requires substantial resources and time. The *τ-optimization* algorithm we developed for training our two first methods, allowed exploring the optimal upper bound of their performance, and the behavior after training with various training set sizes.

The cornerstone of this contribution is a novel iterative approach of the Wikification task, not explored by previous work. This technique, achieves convergence

through a series of iterative steps, each of which aims at improving approximation of the optimal annotations.

The development of a better relatedness measure for Wikipedia entities, based on both incoming links, and contextual content is included among our future plans. We also aim at further improving, fine tuning our methods performance and engineering their large scale deployment as an online service, after reserving the necessary resources.

Acknowledgments

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thales. Investing in knowledge society through the European Social Fund"

References.

1. Cucerzan, S. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. InProceedings of EMNLP-CoNLL 2007: 708-716.
2. Fellbaum, C., editor. WordNet, an electronic lexical database. The MIT Press.1998
3. Ferragina, P., and Scaiella, U. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM '10). ACM, New York, NY, USA, 1625-1628, 2010.
4. Gabrilovich, E., Markovitch, S. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI'07), Rajeev Sangal, Harish Mehta, and R. K. Bagga (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1606-1611, 2007.
5. Han, X., Sun, L., and Zhao, J. Collective entity linking in web text: a graph-based method. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '11). ACM, New York, NY, USA, 765-774, 2011.
6. Han, X., Zhao, J. Named entity disambiguation by leveraging wikipedia semantic knowledge. In Proceedings of the 18th ACM conference on Information and knowledge management (CIKM '09). ACM, New York, NY, USA, 215-224, 2009.
7. Hoffart, J., Suchanek, F.M., Berberich, K., Lewis-Kelham, E., Melo, G., and Weikum, G. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In Proceedings of the 20th international conference companion on World wide web (WWW '11). 2011, ACM, New York, NY, USA, 229-232
8. Hoffart, J., Yosef, M.A., Bordino, I, Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, S. Robust disambiguation of named entities in text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11). Association for Computational Linguistics, Stroudsburg, PA, USA, 782-792, 2011.
9. Kulkarni, S., Singh, A., Ramakrishnan, G., and. Chakrabarti, S. Collective annotation of Wikipedia entities in web text. In Proceedings of the 15th ACM SIGKDD international

conference on Knowledge discovery and data mining. ACM, New York, NY, USA, 457–466, 2009.
10. Makris C., Plegas Y., Theodoridis E., Improved Text Annotation with Wikipedia Entities. In Proceedings of the 28th Annual ACM Symposium on Applied Computing. ACM, New York, NY, USA, 288-295, 2013.
11. Meij, E., Weerkamp, W., and Rijke, M. Adding semantics to microblog posts. In Proceedings of the 5th ACM international conference on Web Search and Data Mining (WSDM '12). ACM, New York, NY, USA, 536-572, 2012
12. Mihalcea, R., and Csomai, A. Wikify!: linking documents to encyclopedic knowledge. In Proceedings of the 16th ACM international Conference on Information and Knowledge Management (CIKM '07). ACM, New York, NY, USA, 233-242, 2007
13. Milne, D. and Witten, I.H. Learning to link with Wikipedia. In Proceedings of the 17th ACM conference on Information and knowledge management (CIKM '08), ACM, New York, NY, USA, 509-518, 2008.
14. Navigli, R. Word Sense Disambiguation. ACM Computing Surveys, 41(2), 10:1-10:69, 2003.
15. Ratinov, L., Roth, D., Downey, D. and Anderson, M. Local and global algorithms for disambiguation to Wikipedia. Volume 1 (HLT '11)