Identifying Topics in Social Media Posts using DBpedia

Óscar Muñoz-García[1], Andrés García-Silva[2], Óscar Corcho[3], Manuel de la Higuera Hernández [4], Carlos Navarro[5]

[1,4,5]Havas Media (Madrid, Spain); [2,3]Ontology Engineering Group (UPM)

E-mail: [1,4,5]{oscar.munoz,manuel.higuera,carlos.navarro}@havasmedia.com,
[2,3]{hgarcia,ocorcho}@fi.upm.es

Abstract: **This paper describes a method for identifying topics in text published in social media, by applying topic recognition techniques that exploit DBpedia. We evaluate such method for social media in Spanish and we provide the results of the evaluation performed.**

Keywords: topic identification, dbpedia

1 INTRODUCTION

Topic recognition (a.k.a. *topic identification*) refers to the task of identifying the central ideas in a text [17]. In the context of social media, topic recognition may be useful for many different purposes, such as automatically summarising the content published in a channel, mining the interests of a given user, etc.

In the context of communication and advertising companies, topic recognition in social media posts can provide several benefits, increasing the effectiveness of the investment in social media advertising which suffers from a significant degree of inefficiency [9]. We envision that automatic topic identification will lead to an efficient investment in media advertising, since advertisement actions will be focussed in the appropriate channels and directed to the most suitable set of users.

Despite that some efforts have been done to structure social media information, such as Twitlogic [26], there is still the need for approaches able to cope with the different channels in the social web and with the challenges they pose. Social media posts are characterised by containing text that varies in length from short sentences in microblogs to medium-size articles in web logs. Very often, text published in social media contains misspellings, is completely written in uppercase or lowercase letters, or it is composed of set phrases, what leads to incorrectly identified topics. As an example, for the Spanish language, the absence of an accent in a word may give such word a completely different meaning. For such case, it is very important for the topic identification method to take into account the context of the post.

In this paper, we present a method that combines NLP (natural language processing), tag-based and semantic-based techniques for identifying the topics in posts published in social media. Such method exploits the semantics of the resources published in the web of data [2]. More specifically, the method makes use of DBpedia [3], a semantic representation of part of Wikipedia information.

Therefore, the topics identified by our method are expressed in terms of DBpedia resources. We consider that DBpedia resources are a good starting point to define keyword meanings due to the fact that a huge part of the knowledge base is related to classes in the DBpedia Ontology. Moreover, currently the DBpedia ontology has 1,667,000 instances. In addition DBpedia resources are linked to other linked data sources and ontologies such as Geonames [13], YAGO [27], OpenCyc [22], and WordNet [8], providing more semantic information in the form of relations such as *typeOf* and *sameAs*. Therefore by linking social media with DBpedia resources we can profit not only from the DBpedia Ontology and the knowledge base facts but also from the interlinked semantic information.

The paper is structured as follows: section 2 summarises previous related work; section 3 describes the method proposed in this paper; section 4 describes the results of the experimentation we have performed; finally, section 5 presents the conclusions and depicts future lines of work.

2 RELATED WORK

Wikipedia is a valuable source of knowledge used in many research tasks due to the wealth and quality of textual information contained on it. Wikipedia content is broad and multilingual. The encyclopaedia is maintained by thousands of editors, and thus it evolves and adapts as events and knowledge change [28]. Articles have been used as concepts and individuals describing things of real life. For instance DBpedia [3] and YAGO [27] are efforts aiming at structuring Wikipedia information using RDF triples. Both projects classify their entities using an Ontology. DBpedia team has created manually an ontology based on infoboxes information while YAGO's has created an ontology based on the Wikipedia hierarchy of categories and Wordnet [8] information.

Textual information as well as the graph of hyperlinks inside Wikipedia has been exploited to extract candidate senses or concepts in different forms. A simple technique described in [25] relates words inside a document to

Corresponding author: Óscar Muñoz-García, Havas Media, Avda. General Perón 38 – 28020 (Madrid), +34 638516756, oscar.munoz@havasmedia.com

Wikipedia articles using just the article title information. Wikipedia links, which associate an anchor text with an article, have been used to collect senses for phrases. Senses are defined as the linked articles, and phrases are the anchor text [5], [20], and [19]. The whole set of Wikipedia articles are considered candidate senses or concepts in [28] and [11]. A graph where nodes are articles and categories and edges represent relatedness between articles has been used in [5] and [28]. Finally some authors propose to take advantage of disambiguation pages and redirection links to select candidate senses and alternative labels respectively [18].

Many natural language approaches tap into Wikipedia information to achieve their goals including (1) text categorization and topic recognition [4], [5], [10], [28], and [25], (2) semantic relatedness between fragments of text [11], (3) keyword extraction [20], and (4) word sense disambiguation [19] among others.

In particular, categories and topics have been vaguely distinguished in the literature and often are treated as synonyms. In our analysis we have distinguished topic recognition and text categorization approaches using supervised learning [4], [10], and [18] from those using unsupervised techniques [5], [28], and [25]. Some other authors have tried both supervised and unsupervised techniques such as the work presented in [20].

Similarly to [18] we use as candidate senses for ambiguous term information taken from disambiguation pages as well as redirection links for alternative labels. However in contrast to this approach ours is unsupervised. With respect to unsupervised approaches we use a vector space model [23] to represent our candidate senses. On the contrary [5], and [28] use a graph. The vector space model has been used in [20], and [25]. However, in [25] authors use just information from the article titles and in [20] candidate senses are collected from hyperlinks in Wikipedia. In short we expect that using human knowledge encoded in disambiguation pages with respect to the possible meanings of a phrase leads to more accurate results since we do not have to decide over the whole set of Wikipedia concepts. In addition, our unsupervised technique learns from Wikipedia corpus avoiding the need of training data that is difficult to gather in environments such as the social web where the vocabulary is in constant change.

The main benefit of our approach in comparison with previous ones that use Wikipedia for topic identification is the interlinking of social media posts with the Web of data, through DBpedia semantic resources.

3 DESCRIPTION OF THE METHOD

The method receives the text of a post *p* and a language *l* as an input and returns a set of topics T_p^l that have been mentioned in *p*. To do so, a pipeline is executed. Such pipeline consists in the ordered execution of the processes detailed next.

3.1 Part-of-speech tagging

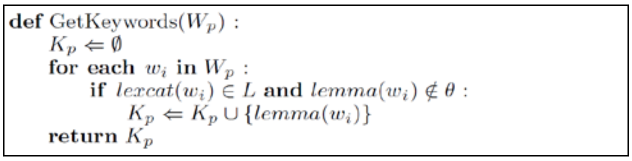
This process takes the text of *p* as an input and returns a set of keywords K_p that appear in such text. For doing so, we apply NLP techniques to annotate each of the words (or lexical units) contained in the post with a lexical category (e.g. *noun*, *verb*, *adjective*). In our work, we filter those lexical units that refer to fixed entities with meaning. More specifically, we only consider those words whose lexical category is one of the following: *common noun*, *proper noun*, *acronym*, *foreign word* and *unit of measure*.

In addition, in this process, each lexical unit is annotated with its lemma. A lemma is the canonical form of a lexeme. A lexeme in morphology refers to the set of all the forms that have the same meaning. Thus, the lemma is chosen by convention from all the items contained in the set lexeme. As an example, *media* and *medium* are forms of the same lexeme, with *medium* as the lemma.

Given the text of the post *p*, we define $W_p = w_1, w_2, ..., w_n$ as the sequence of lexical units contained in *p*. We also define *lexcat(w)* as the lexical category of the lexical unit *w*, and *lemma(w)* as the lemma of *w*. In addition, we define *L* as the set of lexical categories that we consider. Finally, θ is defined as a set of stop words (i.e., lemmas that will be excluded from been inserted in K_p).

Then, our part-of-speech (POS) tagging process consists in the execution of listing 1. Note that the POS tagging process do not differ from previous approaches and it is described for self-containment purposes.

Listing 1: Definition of the POS Tagging process



```
def GetKeywords($W_p$) :
    $K_p \Leftarrow \emptyset$
    for each $w_i$ in $W_p$ :
        if $lexcat(w_i) \in L$ and $lemma(w_i) \notin \theta$ :
            $K_p \Leftarrow K_p \cup \{lemma(w_i)\}$
    return $K_p$
```

In our implementation, we have defined an annotation pipeline with Gate [6] for handling the overall NLP process, while the component in charge of performing the POS tagging is TreeTagger [24].

3.2 Topic Recognition

This process receives the list of keywords K_p and returns the set of topics T_p, as semantic entities derived from K_p.

Once we have spotted the keywords appearing within the analysed text our next step consists of identifying their meaning. We propose to carry out this task by linking each keyword with a ranked list of DBpedia resources.

Note that keywords can be ambiguous and thus the linking to a DBpedia resource is not just the result of a simple string matching process between the keyword and the resource name. For instance, according to Wikipedia [29] the blackberry term refers to a shrub and its fruit, an island, a smart phone, and to a song among others entities.

To carry out this mapping we use Sem4Tags, a technique previously introduced in [12]. Sem4Tags is a configurable process aiming at choosing the semantic entity that better defines the keyword meaning in the context where it is used. By context we mean here the set of keywords identified for the analysed text. In the following we first discuss the concept of context and then we present our disambiguation strategies.

3.2.1 Context

Humans are able to distinguish the sense of an ambiguous word within a sentence due to the understanding of the context where the word appears. By context we understand other words appearing in the same or a neighbour sentence, or in the whole document. However, not all the words in the context help to disambiguate the meaning of a word. According to an early experiment presented in [15], 4 is the number of words above which the context does not add more resolving power to the disambiguation. Let us analyse the following paragraph extracted from a technical forum:

But a hardware problem is more likely, especially if you use the phone a lot while eating. The Blackberry's tiny trackball could be suffering the same accumulation of gunk and grime that can plague a computer mouse that still uses a rubber ball on the underside to roll around the desk.

In this text fragment we can see that words such as *hardware*, *phone*, and *trackball* are more related to the word *Blackberry* than other words such as *gunk*, *grime*, and *rubber ball*. Our hypothesis is that a subset of the most related words to the ambiguous word in the context will produce better disambiguation results than using the whole context. We call this subset the active context.

Table 1: Active context selection for *blackberry* keyword

Keyword	Relatedness	Keyword	Relatedness
phone	0.357	*hardware*	0.347
trackball	0.311	*mouse*	0.311
computer	0.288	*desk*	0.287
problem	0.246	*rubber ball*	0.246
grime	0.190	*gunk*	0.168

To carry out this selection we use a technique described in [14]. After removing repeated words and stop words from the context, we compute the semantic relatedness between each context word and the word to disambiguate. This relatedness computation is performed by using a web-based relatedness measure taking into account the co-occurrence of words on web pages, according to frequency counts, and giving a value between 0 and 1, which indicates the degree of semantic relatedness that holds between the compared words. Finally, we construct the active context set with the context words whose relatedness scores above a certain threshold.

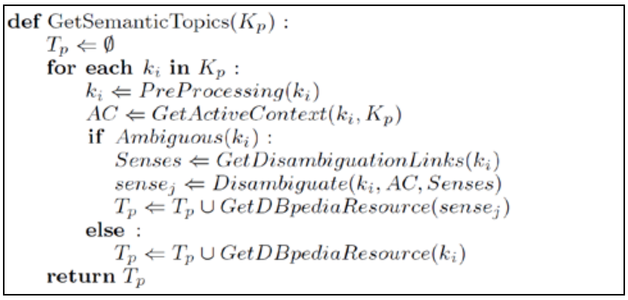
In table 1 different relatedness values calculated between Blackberry and the keywords found in the previous example are displayed. By choosing the top 4 most related keywords we state that the active context for Blackberry consists of the words phone, hardware, trackball, and mouse. We use this active context in our disambiguation task.

3.2.2 Disambiguation

Our process relies on Wikipedia redirection and disambiguation pages. The former are links between alternate titles and an article while the latter are lists of candidate articles defining the possible senses of an ambiguous term. DBpedia currently contains statements formalising redirection and disambiguation pages for the English version of Wikipedia. However for Spanish they are not providing this information [1]. Therefore we harvested links from redirection and disambiguation pages from the Spanish Wikipedia version.

Sem4Tags, see listing 2, pre-processes the keyword to find a normalised representation based on Wikipedia article titles. We benefit from Wikipedia redirection pages when the keyword has been considered as an alternative to an article title. In addition, we modify morphologically the keyword according to the article title notation. Finally, if after those modifications we have not found a Wikipedia article, we use the Yahoo! spelling and suggestion service [31] to find an alternative representation.

Listing 2: Definition of the Topic Recognition process



```
def GetSemanticTopics($K_p$) :
    $T_p \Leftarrow \emptyset$
    for each $k_i$ in $K_p$ :
        $k_i \Leftarrow PreProcessing(k_i)$
        $AC \Leftarrow GetActiveContext(k_i, K_p)$
        if $Ambiguous(k_i)$ :
            $Senses \Leftarrow GetDisambiguationLinks(k_i)$
            $sense_j \Leftarrow Disambiguate(k_i, AC, Senses)$
            $T_p \Leftarrow T_p \cup GetDBpediaResource(sense_j)$
        else :
            $T_p \Leftarrow T_p \cup GetDBpediaResource(k_i)$
    return $T_p$
```

For instance, a keyword such as cell phone is transformed into *Cell_Phone* and then into *Mobile_phone* due to a redirection link between both terms [30]. *Mobile_phone* is actually the article defining the meaning of the keyword in Wikipedia. In DBpedia, *dbpedia:Mobile_phone* is the semantic entity contained in the statements with information about cell phones, where the prefix “*dbpedia:*” stands for DBpedia namespace[2]. Note that the goal of this pre-processing task is to find an equivalent notation in Wikipedia. However for ambiguous keywords this mapping can produce wrong results.

[1] As of January 2011 there are not data sets for redirection and disambiguation links in Spanish [7]

[2] http://dbpedia.org/resource/Mobile_phone

Next, we look for a disambiguation page containing the keyword normalised version. If such disambiguation page does not exist we conclude that the keyword is not ambiguous and return the DBpedia resource related to the Wikipedia article title. On the other hand, if we found a disambiguation page we use the set of links as candidate senses for the keyword. In this last case we need a disambiguation task to select the most appropriate sense for the keyword.

We have two different alternatives to disambiguate the meaning of a keyword. On the one hand we can assign the most frequent sense for the ambiguous word. Wikipedia editors agree on what is the most frequent sense of word and the corresponding article is displayed first when someone poses a query with that word. Despite this strategy seems naive it is very effective. In [21] authors have achieved a 78.89% precision in a disambiguation task when they have used the most frequent sense defined in Wordnet. In our running example the default sense for blackberry would correspond to the fruit.

On the other hand, following Lesk's idea [16] we can measure how similar is the definition of each sense with respect to the context of the ambiguous word, and then select the most similar sense. To do so we use a model where each of the candidate senses (i.e., Wikipedia articles) as well as the keyword and its context are represented as a vector. First we create the *Vocabulary* set as the union of the top *N* frequent terms in each of the candidate senses. Next, for each sense we create a vector in $R^{|Vocabulary|}$ where each position corresponds to an element in an ordered version of the *Vocabulary* set. The value w_i associated with the i-th position in the vector is calculated using TF-IDF[3] [1] for the corresponding i-th term in the ordered set. Note that IDF is calculated only in the set of candidate senses.

Table 2: Disambiguation results for *Blackberry* keyword

DBpedia resource	**Definition**	**Similarity**
BlackBerry	*is a line of mobile e-mail* *and smartphone*	0.224
BlackBerry	*is an edible fruit*	0.15
BlackBerry_(song)	*is a song by the Black* *Crowes*	0.0
Blackberry_Township, *_Itasca_County,* *_Minnesota*	*is a township in ...* *Itasca County*	0.0

Similarly, we create a vector for the keyword and its context. In this case, w_i takes as value 1 if the i-th term in the ordered set appears in the keyword context, and 0 if not. We compare the keyword vector and each one of the sense vectors using as similarity measure the cosine function. Thus, we select the sense vector with the highest similarity value with respect to the keyword vector.

In table 2 we show the similarity values between *Blackberry* and its active context and some of the candidate meanings extracted from Wikipedia. Thus, we select the hand held device (*dbpedia:BlackBerry*) as the one representing the keyword meaning.

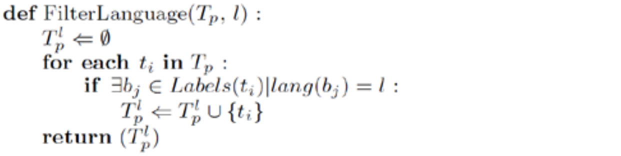
3.3 Language Filtering

This process receives the list of topics T_p and returns the set of topics T^l_p that have been defined for a language *l*.

Given a RDF resource *r*, we define *Labels(r)* as the function that retrieves the values of the literals e that entail the statement <*t*> *rdfs:label* <*e*>. We define *lang(e)* as the function that retrieves the language in which a literal e is expressed.

Given a language *l*, we define the language filtering process consist in the execution of listing 3.

Listing 3: Definition of the Language filtering process



```
def FilterLanguage($T_p$, $l$) :
    $T^l_p \Leftarrow \emptyset$
    for each $t_i$ in $T_p$ :
        if $\exists b_j \in Labels(t_i) | lang(b_j) = l$ :
            $T^l_p \Leftarrow T^l_p \cup \{t_i\}$
    return ($T^l_p$)
```

4 EVALUATION

We have evaluated our method with a corpora of 10,000 posts. Such corpora have been gathered by crawling posts related with the telecommunications domain from the following kinds of media channels:

- Web logs. We have extracted the texts of the posts from the feeds of blog publishing platforms such as Wordpress and Blogger.
- Forums. We have scrapped the text of the comments published in web forums constructed with vBulletin and phpBB technologies.
- Microblogs (e.g., Twitter and Tumbler). We have extracted the short messages published in such channels by querying their APIs.
- Social networks (e.g., Facebook, MySpace, LinkedIn and Xing). We have extracted the messages published in such channels by querying their APIs.
- Review sites (e.g., Ciao and Dooyoo). We have scrapped the text of the comments published in such channels.
- Audiovisual content publishing sites (e.g., YouTube and Flickr). We have extracted the textual comments associated to the audiovisual content.
- News publishing sites. We have extracted the articles from the feeds published in such channels.

[3] TF-IDF stands for Term Frequency and Inverse Document Frequency

- Other sites not classified in the categories above (e.g., Content Management Systems) that publish their content as feeds, or that have a known HTML structure from which a scrapping technique can be applied.

We have measured how the method performs with three variants of the topic identification algorithm. The first variant consists in identifying the topics without considering any context. Thus, we are always assigning to keywords the sense that Wikipedia editors have defined as the default sense for that word. The second variant consists in identifying the topics by considering as context the other keywords found in the same media post. The third variant consists in identifying the topics by applying the active context selection technique.

Table 3 shows the coverage of the processes involved in our method. Row 2 reflects the coverage of the POS tagging process (i.e., the percentages of posts for which at least one keyword has been found). Rows 4-6 show the coverage of the topic identification process (i.e., the percentages of posts for which at least one DBpedia resource has been identified). Rows 8-10 show coverage of the language filtering process (i.e., the percentage of the posts for which at least one DBpedia resource with a Spanish label have been found).

The coverage of the POS tagging process is nearly 100% for all the channels while the coverage of the topic recognition process is over 90% for almost all the cases. However, when the language filtering process is applied, the coverage of the topic recognition is reduced in about 10 points because no all the DBpedia resources are labelled with a Spanish term. In special, the coverage of the topic recognition process for the review sites is less than for the rest of channels. The reason for that is that, in this kind of sites, there is information about specific product models whose commercial denomination is not necessarily translated to a Spanish label. In general, the coverage for web logs and news publishing sites is the highest. The reason for that is that the length of the posts published in such channels is greater than in the other channels. In general, if context is taken into account, the coverage of the method is bigger.

We have evaluated the precision of our method with a random sample of 1,816 posts (18.16% of the evaluation set) using 47 human evaluators. We have shown each post, the topics identified to three different evaluators. For each topic, the evaluators have selected one of the following options: (1) the topic is not related with the post, (2) the topic is somehow related with the post, (3) the topic is closely related with the post, or (4) the evaluator has not enough information for taking a decision. We applied Fleiss' kappa test to measure the agreement among the evaluators. The strength of agreement for 2 evaluators is very good (0.826). Such strength is moderate (0.493), if 3 evaluators must agree on the same answer. We have considered an answer valid if at least two evaluators agree on it.

Table 4 shows the results of the evaluation of the precision. The precision of the topic identification process depends on the channel and its value range from 59.19% for social networks to 88.89% for review sites. One of the reasons that explain such variability is the specificity of the keywords included in the posts of the different channels. As an example, in review sites the posts use to include references to specific brands or models, while in social networks such references are more ambiguous. Another reason is that some channels (e.g., social networks) include more misspellings than other channels (e.g., news publishing sites). With respect to the precision obtained by considering the context or not, there is not a general rule. While the first variant (without context) provide a better precision in most of the cases, the second variant (considering the other keywords in the post as context) is better for web logs, and the third variant (active context) is better for microblogs and review sites.

5 CONCLUSIONS

In this paper we have described a method for identifying topics in social media posts using DBpedia. The method consists in the execution of a pipeline of four consecutive processes. Firstly, a NLP process is executed to perform the morphological analysis of the text contained in the post. Secondly, we apply a topic recognition process for identifying the topics contained in the text. Finally, a language filtering process is executed to filter the topics that are not labelled with terms expressed in a given language.

We have achieved good results of coverage. The precision of the method depends on the social media channel and with respect to considering context or not, there is not a

Table 3: Coverage of the method

	Blogs	Forums	Microblogs	Soc. N.	Others	Reviews	Audiovisual	News	All
POS Tagged	99.63%	96.64%	99.01%	98.14%	98.77%	98.53%	97.20%	99.52%	98.38%
Topic Recognition									
Without context	96.7%	87.68%	94.22%	93.54%	92.71%	88.81%	90.29%	96.67%	92.35%
With context	96.64%	93.07%	95.54%	94.99%	95.13%	92.67%	97.41%	98.54%	95.02%
Active context	99.24%	89.71%	94.43%	96.4%	94.75%	93.81%	92.23%	97.4%	94.72%
Topic Recognition (after language filtering)									
Without context	91.21%	79.04%	87.54%	82.64%	86.93%	70.15%	82.52%	90.71%	82.74%
With context	88.43%	80.84%	86.31%	85.24%	88.72%	76.19%	89.66%	92.46%	84.85%
Active context	89.69%	80.51%	86.51%	86.78%	89.78%	75.59%	80.58%	90.54%	84.73%

Table 4: Precision of the method

	Blogs	Forums	Microblogs	Soc. N.	Others	Reviews	Audiovisual	News	All
Without context	67.48%	66.67%	59.72%	72.32%	59.19%	79.17%	84.44%	71.95%	68.42%
With context	75.61%	59.35%	54.88%	65.71%	53.52%	83.87%	77.78%	64.37%	63.11%
Active context	67.71%	64.45%	65.58%	70.1%	49.15%	88.89%	79.07%	71.93%	66.59%

variant that provide the best results for all the channels.

Some social media channels are characterised by containing text with variant quality from an orthographical and grammatical perspective. Misspelled text often leads to incorrectly identified topics. We use Yahoo! spelling web service for solving some misspellings. Nevertheless, the overall precision results can be improved by dealing with different kinds of slang (e.g., certain expressions by teenagers). In addition, when keywords are extracted from set phrases, the topics identified do not reflect the sense of the post. In order to improve the precision, the NLP process must be able to detect such set phrases and to exclude them from the text analysed.

Acknowledgments

This work has been supported by the CENIT project Social Media (CEN-20101037) and a FPI grant (BES-2008-007622) of the Spanish Ministry of Science and Innovation. We are very grateful to all the evaluators from UPM and Havas Media.

References

[1] R. Baeza-Yates, B. Ribeiro-Neto, et al. Modern information retrieval. Addison-Wesley Harlow, 1999.

[2] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. Int. Journal on Semantic Web and Information Systems (IJSWIS), 2009.

[3] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - A crystallization point for the Web of Data. Journal of Web Semantics: Science, Services and Agents on the WWW, 7:154-165, September 2009.

[4] Z. Bodo, Z. Minier, and L. Csato. Text categorization experiments using Wikipedia. In Proc. of the Int. Conference on Knowledge Engineering, Principles and Techniques, 2007.

[5] K. Coursey, R. Mihalcea, and W. Moen. Using encyclopedic knowledge for automatic topic identification. In Proc. of the Thirteenth Conference on Computational Natural Language Learning, pages 210-218. Association for Computational Linguistics, 2009.

[6] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics, 2002.

[7] DBpedia. Downloads page. http://wiki.dbpedia.org/Downloads.

[8] C. Fellbaum. WordNet: An Electronical Lexical Database. The MIT Press, Cambridge, MA, 1998.

[9] Forrester. Media Buying Goes Real Time. How dynamic optimization of online media buying will affect advertisers, publishers, agencies and ad networks. Report, February 2010.

[10] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In Proc. of the National Conference on Artificial Intelligence, volume 21, page 1301. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[11] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In Proc. of the 20th Int. Joint Conference on Artificial Intelligence, pages 6-12, 2007.

[12] A. García-Silva, O. Corcho, and J. Gracia. Associating semantics to multilingual tags in folksonomies (poster). In 17th Int. Conference on Knowledge Engineering and Knowledge Management EKAW 2010 (poster), Lisbon (Portugal), October 2010.

[13] GeoNames. GeoNames geographical database, 2010. Last access on Jan 2011 at: http://www.geonames.org/export.

[14] J. Gracia and E. Mena. Multiontology semantic disambiguation in unstructured web contexts. In Proc. of Workshop on Collective Knowledge Capturing and Representation (CKCaR'09) at K-CAP'09, Redondo Beach, California (USA). CEUR-WS, ISSN 1613-0073, September 2009.

[15] A. Kaplan. An experimental study of ambiguity and context. Mechanical Translation, 2:39-46, 1955.

[16] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In Proc. of the 5th annual Int. conference on Systems documentation, SIGDOC '86, pages 24-26, New York, NY, USA, 1986. ACM.

[17] C. Y. Lin. Knowledge-based automatic topic identification. In Proc. of the 33rd annual meeting on Association for Computational Linguistics, pages 308-310, Morristown, NJ, USA, 1995. Association for Computational Linguistics.

[18] O. Medelyan, I. Witten, and D. Milne. Topic indexing with Wikipedia. In Proc. of the AAAI WikiAI workshop, 2008

[19] R. Mihalcea. Using wikipedia for automatic word sense disambiguation. In Proc. of NAACL HLT, 2007.

[20] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In CIKM, volume 7, pages 233-242, 2007.

[21] R. Navigli, K. C. Litkowski, and O. Hargraves. Semeval-2007 task 07: Coarse-grained English all-words task. In Proc. of the Fourth Int. Workshop on Semantic Evaluations (SemEval-2007), pages 30-35, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[22] S. L. Reed and D. B. Lenat. Mapping ontologies into cyc. Technical report, Cycorp, Inc" 2002.

[23] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. Commun. ACM, 18:613-620, November 1975.

[24] H. Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In Proc. of the Int. Conference on New Methods in Language Processing, pages 44-49, 1994.

[25] P. Schonhofen. Identifying document topics using the Wikipedia category network. Web Intelligence and Agent Systems, 7(2):195{207, 2009.

[26] J. Shinavier. Real-time #semanticweb in <= 140 chars. In Proc. of the Linked Data on the Web Workshop (LDOW2010), Raleigh, North Carolina, USA, April 2010.

[27] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A Large Ontology from Wikipedia and WordNet. Elsevier Journal of Web Semantics, 2008.

[28] Z. Syed, T. Finin, and A. Joshi. Wikipedia as an ontology for describing documents. In Proc. of the Second Int. Conference on Weblogs and Social Media. AAAI Press, 2008.

[29] Wikipedia. Blackberry disambiguation page. http://en.wikipedia.org/wiki/Blackberry_(disambiguation).

[30] Wikipedia. Cell phone redirection page. http://en.wikipedia.org/w/index.php?title=Cell_Phone&redirect=no.

[31] Yahoo! Spelling Suggestion Web Service. http://developer.yahoo.com/search/web/V1/spellingSuggestion.html