Short Text Categorization Exploiting Contextual Enrichment and External Knowledge

Stefano Mizzaro, Marco Pavan, Ivan Scagnetto, Martino Valenti[*]
Dept. of Mathematics and Computer Science - University of Udine
via delle Scienze, 206
Udine, Italy
{mizzaro, marco.pavan, ivan.scagnetto}@uniud.it, martino.valenti@gmail.com

ABSTRACT

We address the problem of the categorization of short texts, like those posted by users on social networks and microblogging platforms. We specifically focus on Twitter. Since short texts do not provide sufficient word occurrences, and they often contain abbreviations and acronyms, traditional classification methods such as "Bag-of-Words" have limitations. Our proposed method enriches the original text with a new set of words, to add more semantic value by using information extracted from webpages of the same temporal context. Then we use those words to query Wikipedia, as an external knowledge base, with the final goal to categorize the original text using a predefined set of Wikipedia categories. We also present a first experimental evaluation that confirms the effectiveness of the algorithm design and implementation choices, highlighting some critical issues with short texts.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms; Experimentation

Keywords

context-aware retrieval, enrichment, wikipedia, evaluation

1. INTRODUCTION

Social media are widely used nowadays, also thanks to the wide spread of mobile devices which allow users to post contents from everywhere and at every time. The well known Twitter platform allows users to write and share short texts with a limited length (140 characters). This restriction, combined with a very frequent quick writing activity carried out by moving users, often with pervasive abbreviations and new coined acronyms, opens new challenges to text categorization systems. In most of cases, short texts do not have enough words to provide sufficient information for both topic detection and text classification tasks. Also, abbreviations, acronyms and even new formulated words make harder the process of extracting information. In some cases there is an even more serious problem, due to grammar mistakes, misspellings or typos, which complicates the interpretation of text.

A common representation of a text or document uses the "Bag-of-Words" model, and Information Retrieval techniques can be applied in order to evaluate how important a word is, in that document. Several works in the literature exploit external sources to enrich the original set of words with other additional words, in order to add semantic value to the text and improve the categorization process; they will be described in the next section of this paper. Many of these approaches [1, 3, 4, 5, 6, 8, 10, 11] focus on semantic and syntactic analysis in order to better detect the meaning of words and phrases by solving problems such as synonymy or polisemy, but giving not so much importance to the temporal context, i.e. *when* a sentence is made. Topics and concepts expressed in short texts on social networks are often strictly related to the temporal context [2] in which they are posted. Same expressions or set of words can be referred to different topics if posted in different moments. For instance, an exultation of victory for a sports competition depends on the game played one or few days before, or an opinion about a topic is related to recent news on TV or on the Web; therefore, in any case, not too distant in terms of time. Also, it is possible to exploit the additional features offered by the social media platforms, such as hashtags, mentions, or directly a link embedded into the posted text, in order to have more information for the categorization. But this technique fails when users post plain text, with just simple words; therefore, another approach is needed in order to have an enrichment process that works in any case.

On this basis we propose a prototype for categorization of short texts (e.g., texts posted on Twitter). The final outcome of our system is the assignment of a (very short) list of labels extracted from Wikipedia categories, exploiting their relationships whithin the *Category Tree*, to short texts posted on Twitter. To improve the process, our novel proposal provides a module which analyzes the text, searches the Web for related documents, and extracts a set of words in order to enrich the original text with additional semantic value.

In Section 4 we describe the experimental evaluation of our approach which is essentially test collection based, following the principles of TREC [9]. We focus on texts posted on Twitter because of its popularity and of its very strict length limitation (140 characters). Thus, we are sure to test the contextual enrichment approach in an interesting and difficult case.

The paper is organized as follows. Section 2 summarizes some related work about short text categorization, enrichment approaches,

---

[*]Authors are listed alphabetically

and external knowledge sources. We then present the details of our proposed algorithm describing its aims and implementation (Section 3), and its experimental evaluation (Section 4). We discuss the results in Section 5, and we draw some conclusions and describe some future work directions in Section 6.

2. RELATED WORK

Recently, there has been a growing interest among researchers on how to categorize and classify short texts, due to the wide spread of social media platforms, which are an important information source for analysing users' discussions and behaviours. Several works use the clustering approach with the aim of measuring similarity between texts and grouping those that deal with the same topic. Banerjee et al. [1] propose a system for clustering similar items in the feed reader, to make the information more manageable for users, by enriching their representation with additional features from Wikipedia. Also Hu et al. [3] rely on Wikipedia as an external knowledge-base for document clustering, by mapping texts to Wikipedia concepts and categories. In other proposals, Wikipedia is exploited to compute semantic relatedness between words or texts, like in [11], and more recently to identify the word sense with a disambiguation process, as described in [4]. Another recent use of Wikipedia knowledge is to enrich the semantic expression of a target commercial advertisement, as presented by Xu et al. in their work on contextual advertising [10].

Short texts raise new challenges to traditional text mining tasks; therefore, many propose to adopt the enrichment approach to overcome the problems due to insufficient information. Tang et al. [8] propose a framework which performs multi-language knowledge integration for clustering. Sahami et al. [6] address the problem of measuring the similarity of short text snippets by leveraging on web search results, to infer a wider context for each short text (so doing, they can more easily solve ambiguity issues). In a recent paper Meng et al. [5] propose a method to expand short texts with the help of public search engines, by crawling related pages and getting contents as background knowledge of the original short text.

Moreover, the ephemeral nature of Twitter posts begins to suggest to take into consideration the temporal dimension. For instance, Cataldi et al. [2] propose a technique to detect the most emergent topics expressed by the community on Twitter. They consider as emerging a term that frequently occurs in a specified time interval but it is rare in the past, and also take into account the source, by analyzing the author and his social relationships.

The external knowledge and the enrichment process are very useful to understand the meaning of words for categorization, but most of those approaches focus on groups of texts to apply comparisons or clustering. For a single text, poor of information, another approach is needed to select labels that identify the main topic.

3. PROPOSED SYSTEM

The system in this paper aims to categorize short texts by querying the Category Tree of Wikipedia to get a set of labels representing the topics discussed in the text.

Our research and implementation were carried out incrementally, starting from simpler systems and "evolving" them towards more sophisticated solutions. We started by defining a first version of our system, named W2C (i.e., Words to Categories), exploiting only Wikipedia for text labelling. We computed what categories are related to each word in the text, to infer the topics. We then developed two more effective versions, FEL (i.e., Fixed Enrichment and Labelling) and WEL (i.e., Weighted Enrichment and Labelling), which use enrichment techniques to add information to the original

short text, to improve the categorization during the labelling process. The enrichment is carried out with the help of the Google search engine[1]. The three systems are described in full details in the following sections. We kept W2C as a term of comparison for our novel proposals, in order to measure the improvements made by the enrichment process.

3.1 W2C text labelling

The W2C system exploits the relationships within the Category Tree of Wikipedia for extracting the appropriate set of categories for each given short text. We preferred to work with Wikipedia since it is continuously updated with articles about news and popular events (being those the main topics of new tweets). However, the software architecture of our system is source-independent and we could easily switch to other databases (in the same way that we could change the web search engine). Moreover, another reason of our choice is to show how it is possible to exploit Wikipedia in a different way, w.r.t. related works, by using techniques based on the Category Tree.

3.1.1 Step 1: Wikipedia article selection

First, W2C queries Wikipedia APIs with each pair of words (bigram) from the short text. We use bi-grams and not single terms to get a set of articles more homogeneous, and to avoid too much generic articles, due to the words polysemy. With single-term queries it is difficult to focus on one or few topics, and we lose the semantic relations defined by the user who posted the text.

We define Q as the set of queries to perform, with $|Q| = \binom{|A|}{2}$, where A is the set of words extracted from the short text. $\forall a \in A, w(a)$ is the weight of the word in the original text. In this case we always set that weight to 1, for this first version where we do not compute the relevance score of each word. $\forall q \in Q$ we have a query weight defined as follows:

$$w(q) = \sum_{a \in q} w(a). \tag{1}$$

Hence, in this particular case $w = |q|$. By performing this set of queries to Wikipedia, we obtain a set of articles, ranked by the relevance computed by the Wikipedia search engine.[2] For all $q \in Q$, there exists a (possibly empty) set R_q of relevant articles for q. We define $i_q(x) \in [0, |R_q| - 1]$ as the index of each article $x \in R_q$, and then we define the article weight as follows:

$$w_q(x) = \begin{cases} \frac{|R_q| - i_q(x)}{|R_q|} & x \in R_q \\ 0 & \text{otherwise.} \end{cases}$$

We combine all resulting articles in order to obtain a final set X with distinct entries as follows:

$$w(x) = \sum_{q \in Q} w(q) \cdot w_q(x).$$

Therefore, for a query $q \in Q$ and an article $x \in X$, $w_q(x) = 0 \Leftrightarrow x \notin R_q$, hence the query q does not change the final score of x. Also, the higher the number of queries with $x \in R_q$, the higher the weight $w(x)$ will be.

3.1.2 Step 2: Label selection

As second step, in order to have a set of labels to associate with each article extracted during the previous phase of W2C, the

---

[1]However, the system can be easily adapted to use other public search engines.
[2]http://en.wikipedia.org/wiki/Help:
Searching#Search_engine_features

1. Literature	Letteratura
2. Economics	Economia
3. History	Storia
4. Philosophy	Filosofia
5. Science	Scienza
6. Entertainment=(Hobby, Entertainment)	Intrattenimento=(Hobby, Intrattenimento)
7. Finance	Finanza
8. Politics=(Politics, Law)	Politica=(Politica, Diritto)
9. Food and drink	Alimentazione
10. Video games	Videogiochi
11. Computer science	Informatica
12. Health and fitness=(Health, Physical fitness)	Salute e fitness=(Salute, Fitness)
13. Fashion	Moda
14. Medicine	Medicina
15. Music	Musica
16. Engines=(Automobiles, Auto racing, Motorcycle sport)	Motori=(Automobili, Automobilismo, Motociclismo)
17. Photo and Video=(Photography, Film)	Foto e Video=(Fotografia, Cinema)
18. Sports	Sport
19. Places=(Tourism, Geography, Travel)	Luoghi=(Turismo, Geografia, Viaggi)
20. Meteorology	Metereologia

Table 1: Wikipedia categories used in our systems, in English and Italian. The notation X=(Y,Z,...) denotes the labels we made to group categories about related topics.

W2C system extracts a set of Wikipedia categories to assign to the short text. The selection is based on the categories associated to each Wikipedia article selected in the previous step (the categories related to an article are listed at the bottom of every Wkipedia page, as described by Wikipedia guidelines), but we also exploit the Wikipedia category graph. The Wikipedia categories graph is organized so that each category is connected with each of its subcategories; therefore, the distances between nodes also represent the semantic relation values. More precisely, we selected a subset of macro-categories in Italian language to properly classify the tweets of some popular Italian accounts. The corresponding categories are listed in Table 1.

More formally, the second step of W2C is as follows. Let $G = (C, E)$ the categories graph, where $C = \{c_1, c_2, ..., c_n\}$ is the set of all categories, and E the set of directed edges. We say that there exists $e_{c_i, c_j} \in E \Leftrightarrow c_j$ *isSubcategoryOf* c_i. Let $L \subset C$ be the set of macro-categories selected for text categorization, listed previously in Table 1. Let $x \in X$ be an article extracted during the previous phase, we define $C_x \subset C$, the set of categories directly related to the article, as our starting set. Then, for each $c_i \in C_x$, we define $C_{c_i} \subset C$, the set of categories reachable with a path from c_i. We are interested in just few of those, specifically if they are in our selected set (namely, L), therefore $L_{c_i} = C_{c_i} \cap L$.[3] At this point we have restricted L to L_{c_i}, and we denote by l_i the labels extracted form L_{c_i} as follows:

$$l_i = l \in L_{c_i} : sp(l, c_i) = \min_{l \in L_{c_i}} sp(l, c_i),$$

where $sp(l, c_i)$ is the shortest path from l to c_i. The shortest path may not be unique, so there may be more than one l that satisfies the condition. In that case we keep all the retrieved labels. Let L_x the set of $l_i \in L$ selected with this approach, we define the label relevance value as follows:

$$r(l) = n(l) \cdot \frac{1}{\overline{sp(l)}},$$

where $\overline{sp(l)}$ is the mean length of all shortest paths from l to the associated categories, and $n(l)$ the number of these categories. By

---

[3]This set can also be empty. In that case the category c_i does not affect the labels detection.
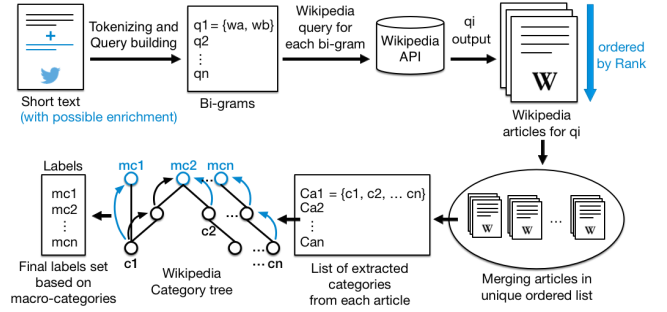

Figure 1: Workflow of the labelling process.

selecting the label with the max $r(l)$ we get the most relevant category for that article,[4] as follows:

$$l_x = l \in L_x : r(l) = \max_{l \in L_x} r(l).$$

By repeating this process for each extracted article we obtain the set L_X of all the labels which potentially represent the topic discussed in the short text. We define a new ranking function for labels to select the most relevant as follows:

$$\forall l \in L_X, r'(l) = \sum_{x \in X_l} w(x),$$

where $X_l \subset X$ is the set of labelled articles with the specific label l, and $w(x)$ the weight of article $x \in X$. With this final ranked list, by selecting the first label, with the highest relevance score, we obtain the topic that is the best match for the analyzed short text. However, we prefer to keep a set of 5 labels (at most), with related relevance scores, in order to analyze eventual subtopics discussed in the text, and to test how precise the proposed system is, in all three versions. Figure 1 shows an overall representation of the labelling process.

3.2 The contextual enrichment approach

To improve the short text categorization, our proposal consists in combining our previously described system, W2C, with an enrichment algorithm, the latter being the Step 0 of our most sophisticated solution. It will be presented in two versions, FEL and WEL, to show different approaches during the final phase, when we select the new set of words to add to the original short text. The enrichment process is the same for both versions, and consists in querying the Google APIs with the short text, in order to get related web pages, with attention on the temporal context. Such pages are then used to infer other terms to add to the original short text. Then, the enriched sentence will be used to query Wikipedia, as described in Section 3.1, with the reasonable hope to obtain a more precise categorization.

Often the texts posted by users on social networks, and in particular on Twitter, are ephemeral and strongly connected with events and news very close to the posting time; therefore, a key feature of our system is to query the web search engine a short time after the text publication. We chose to query Google a few hours after the tweets publication.

For our query q we define $D = \{d_1, d_2, ..., d_n\}$, the set of n retrieved documents,[5] and $K = \{k_1, k_2, ..., k_m\}$, the set of all terms extracted from each $d_i \in D$ (by removing stopwords). We compute the *tf* weighting factor, as usual, for each term for each

---

[4]The label with max value may not be unique. In that case we keep all the labels with max value.

[5]We selected the first 20 documents retrieved by Google, in order to have an adequate number of terms to analyze.

document, but we are interested in how frequent is a word inside the entire collection to understand if the contents are homogeneous in terms of semantics. With this approach we can identify if the original text has meaning, or if it is a set of "random" words, not related with each other or with events or news. To achieve that, we compute the average *tf* vector as follows:

$$tf_i = \frac{1}{n}\sum_{j=1}^{n} d_{ji}^{TF}$$

where d_{ji}^{TF} is the *tf* weighting factor for the term k_i in the document d_j.

We define the relevance score by also considering the document frequency as an indicator of homogeneity, as follows:

$$r_i = tf_i \cdot log(df_i), \forall i \in [1, m].$$

The use of document frequency, in place of inverse document frequency, emphasizes terms that appear in many documents, therefore once again in favor of the homogeneity, that guarantees a meaningful text.

Finally, to refine the ranking function, we tune up the terms weight by considering the word frequency into the corpus of natural language.[6] We define the *it* vector where $\forall it_i$, with $i \in [1, |K|]$, it_i is the frequency of terms k_i into the Italian language corpus.[7] Therefore, the ranking function is $r'_i = r_i - \alpha \cdot it_i$ where $\alpha \in [0, 1]$ is a constant to tune the frequency (we use $\alpha = 0.2$, set empirically). Thus, we get the following ranking function that emphasizes terms if the collection is homogeneous and penalizes very frequent terms:

$$r'_i = tf_i \cdot log(df_i) - \alpha \cdot it_i.$$

3.2.1 FEL - Fixed Enrichment and Labelling

The FEL version of our system (i.e., Fixed Enrichment and Labelling), simply applies a fixed cut-off on the resulting list of terms, computed as previously described. We defined empirically a threshold equal to 5 to set an adequate number of words to use for the next phase in the W2C module. Indeed, during preliminary tests we observed the relevance score distribution of the extracted words, to have an idea of how many words got high scores and could be selected for the enrichment. We chose the first 5 because with lower numbers we lose important words, while considering higher numbers is another case of study, described in the next section as our alternative approach. FEL was developed to test the enrichment effectiveness with no sophisticated cut-off techniques, so that we can see and measure the differences with other cut-off approaches.

3.2.2 WEL - Weighted Enrichment and Labelling

By looking at the final list of terms, and in particular at word scores, we noticed that terms often tend to cluster at the top with similar values, then there is a collapse of the score, which we call "jump", and then, eventually there is another grouping or they go down without a precise rule to the lower values. This score distribution denotes a semantic value for terms that tend to group, i.e. a potential topic represented by that set of words. The WEL version of our proposed system (i.e., Weighted Enrichment and Labelling) takes into account that observation to compute the right threshold to cut-off the final list of terms, by analyzing the differences between the relevance scores of consecutive terms. We define "high jump"

---

[6] Zipf's law states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table.
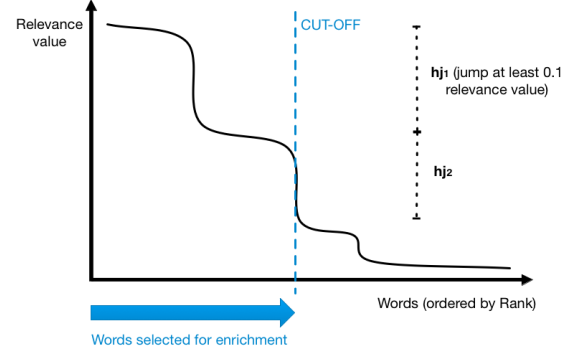
[7] Data extracted from http://http://www.istc.cnr.it/grouppage/DformT.

Figure 2: WEL Cut-off dynamic threshold, based on the difference between words' relevance value

a score difference greater than 0.1,[8] and we seek in the list of terms (ordered by rank) for the second one, in order to keep an amount of words that guarantees a second topic represented by the second group of terms. If the process does not find a second "high jump" we keep just the first one as index for the threshold; also, if none is found, the system sets the threshold equal to 5, as for the FEL version. Figure 2 shows an example of relevance score distribution, with some "jumps" where the words get a much lower score, and the cut-off threshold position that selects the second "high jump".

The WEL dynamic cut-off computation is defined as illustrated in Algorithm 1.

Algorithm 1 Dynamic cut-off threshold

1: $hj_i, i, currentJumps \leftarrow 0$
2: $fixedCutoffThreshold \leftarrow 5$
3: $jumpThreshold \leftarrow 2$
4: $l \leftarrow wordsList.length$ ▶ l is the length of the array containing the words selected for the enrichment process
5: **while** $currentJumps < jumpThreshold$ & $i < l - 1$ **do**
6:     **if** $diffRelevanceForWordsAtIndex(i, i+1) \geq 0.1$ **then**
7:         $currentJumps \leftarrow currentJumps + 1$
8:         $hj_i = i$
9:     **end if**
10:     $i \leftarrow i + 1$
11: **end while**
12: **if** $currentJumps > 0$ **then return** hj_i
13: **else return** $fixedCutoffThreshold$
14: **end if**

As final step for both FEL and WEL systems, the new set of words is sent to the W2C module to start the labelling process and get the set of categories which represent the topics. In both cases words have in general different weights: the W2C module will apply Formula (1) (see Section 3.1).

The phases of the enrichment process previously described are shown in Figure 3.

4. EXPERIMENTAL EVALUATION

We defined a benchmark constituted by three components, as usual in TREC-like IR benchmarks, to test our proposed system. As collection of "documents" we have a set of 20 selected labels extracted from the Wikipedia macro-categories by grouping similar topics (see Table 1); the statements of information needs are 30 short texts extracted from 10 public Twitter accounts dedicated to

---

[8] $hj = 0.1$ is a parameter set empirically, by observing the relevance score distribution.
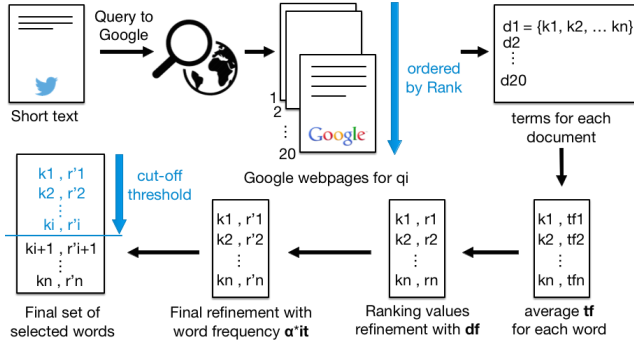
Figure 3: Workflow of the enrichment process.

different topics. We selected very recent tweets, in Italian language, with at least three words; as relevance judgements we have a set of evaluations made by 66 people using a Likert scale on single item. The short texts for the evaluation have been extracted one day before the test session with relevance assessors, by selecting the most recent ones (from 1 to max 5 tweets from each account, starting by the last posted), taking out replies, retweets and parts of conversations. Then, the texts have been submitted to our categorization system after about 6 hours, in order to guarantee the presence of related contents on the web (i.e., the temporal context). We have run all three versions of this system, W2C, FEL and WEL for all the 30 short texts, to get three sets of labels for each text to show to relevance assessors as topics discussed in those texts.

The sample of relevance assessors chosen to perform the test was composed of 66 people, distributed as follows:

- 73% men, 27% women;

- 79% with age between 21 and 30, 21% more than 30;

- 85% with very good familiarity with smartphones;

- 38% with professional knowledge on mobile devices (developers);

The reason for this high number of people and their variety is related to the nature of the analyzed texts; indeed, what people post on social networks often is not easily and uniquely categorizable, and requires the knowledge of related news or events discussed. With this sample we have a good set of ratings which make more reliable the evaluation.

To guide them during the evaluation we have defined a test protocol with detailed instructions, to explain what aspects to take into account for a proper evaluation. We have provided them with an ad-hoc test tool developed for this purpose. It shows first a preliminar page for data gathering about age, sex and how much familiarity they have with smartphones; then, it displays a tweet randomly selected from our set of 30 selected tweets, and a set of labels computed by one of the three versions of our system. As precaution to avoid clues during the test, the set of labels was selected randomly to make not clear to relevance assessors how to associate the algorithms to the corresponding suggestions. The labels inside the set are ordered by relevance (computed by the related algorithm) so that the assessors can understand the accuracy of the system and properly give their evaluation.

For each of the 30 short texts, relevance assessors rate the associated set of labels with a number between 1 and 5 (1=lowest value, 5=highest value) indicating how the labels properly represent the topics discussed in the analyzed text, with attention on how accurate they are. Therefore a greater number of labels indicates noise

during the topic detection, hence a low precision of that algorithm for that specific text. During the test they evaluate each set of labels proposed as a whole (instead of evaluating each single label in the set) with a rating that expresses the global accuracy of the algorithm. We chose this approach, again due to the nature of texts posted on social networks. Most of times, they have complicated language, therefore we focused on the global performance of our algorithms, in term of precision on labelling.

5. RESULTS

5.1 Retrieval effectiveness

To analyze the performance of our proposed system, and the differences between the three versions, the chart in Figure 4 shows the average rating obtained by each version, also with details of all 30 short texts evaluated.

The overall score shows that the FEL version got the best evaluation, although in some cases the other systems got higher ratings. A good point of discussion is the cut-off function; indeed, these results show how the more sofisticated solution WEL got lower ratings than FEL, that uses a simpler cut-off with fixed threshold. During the labels extraction process we have observed that the dynamic cut-off function sometimes introduces too many terms, with the consequence of making difficult the categorization. In particular, by looking at individual texts, FEL has higher effectiveness in most of texts, except for the numbers 4 and 6, where W2C won (see Figure 4). In those cases the low performance is due to an heterogeneous set of words that span to a large number of topics; therefore, the enrichment process adds other unrelated words and makes the original text even more confusing.

The charts in Figure 5 show the rating distribution that relevance assessors have applied to each set of labels they evaluated. It is clear how W2C got more low ratings than FEL and WEL, and how the higher mean and median confirm the superiority of FEL. However, also WEL has led to good improvements over the W2C version, demonstrating how the enrichment process makes more precise the algorithm.

Summarizing, the results show that the FEL solution is more effective and outperforms both WEL and W2C.

5.2 Statistical significance

We have run some statistical tests to determine whether there are any significant differences between the means of relevance judgments got by the three systems. First, we needed to know if the distributions of relevance scores in the datasets were normal, therefore we run a Shapiro-Wilk normality test. The resulting p-value < 0.05, for each group of relevance judgments, indicates a non-normal distribution. With this result a Levene test is needed to verify the homogeneity of variances (as it is usual for this kind of datasets which do not have normal distribution). In this case the p-value was 0.0228, once again smaller than the threshold 0.05, therefore another negative result. As final step we ran the Friedman test, to verify if datasets have significant differences, with a resulting p-value $< 2.2e\text{-}16$. We can observe that this very low p-value is even lower than the threshold 0.0167 obtained by applying the Bonferroni correction; therefore, this output indicates a statistically significant difference between means. At the end, we ran the post-hoc analysis for Friedman's test as implemented in the R system [7], to know which specific groups differed. We obtained a value equal to zero for each pair of datasets; therefore this parameter indicates a significant difference between each proposed version of our system. The W2C has been overperformed by WEL, and even better the FEL solution outperformed both the other versions.
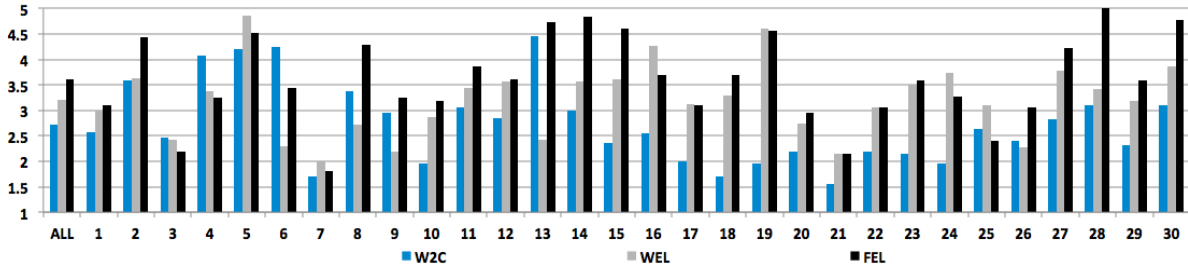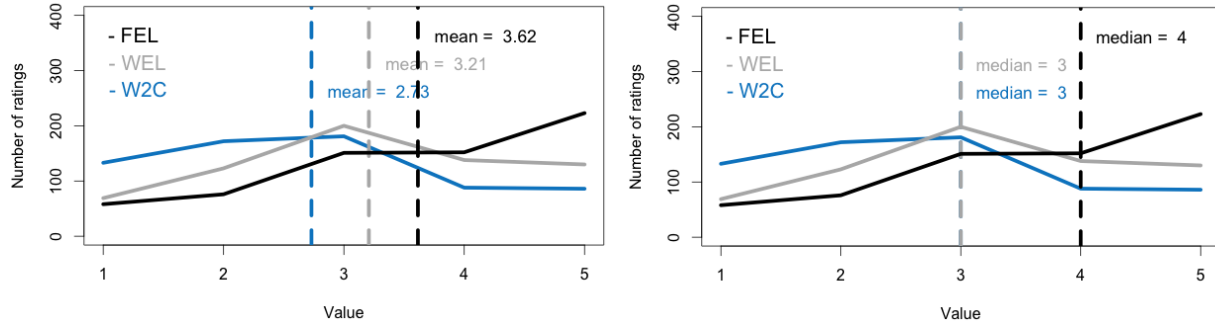
Figure 4: Average rating for each short text

Figure 5: Rating distribution for W2C, WEL, and FEL, with means (left) and medians (right)

6. CONCLUSIONS AND FUTURE WORK

In this paper we have presented our short text categorization system. We developed a first version, W2C, that exploits Wikipedia as external knowledge source. Then, we improved it with an enrichment approach by developing FEL and WEL. The labels proposed by FEL have been evaluated better than the other solutions. In general, the enrichment improved the topic detection, but the cut-off function still needs to be enhanced to better exploit the semantic relations between words in the final rank list. Despite our observations in Section 3.2.2, the parameters used in the algorithm need to be tuned up to refine the cut-off index.

Our system represents a new proposal for short text categorization that does not need the help of URLs inside the text, or hashtags, or other social media features. With this approach it can be used also for general short texts, such as text messages, or vocal messages, on mobile phones. On this basis, we have planned to run other experiments to test new settings for the enrichment process with the goal to better emphasize the semantic relations between extracted words. We can also select different sets of macrocategories from Wikipedia for the W2C module, to test the system with other levels of granularity for topics. Another future work is related to user modelling; we planned to run this system on a set of short texts extracted from a single user social network account. Thus, we can try to detect the main topics discussed by the given user. This work can be a new approach for the development of a new proposal for user modelling based on social data.

7. REFERENCES

[1] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using wikipedia. *SIGIR '07*, July 2007.

[2] M. Cataldi, L. di Caro, and C. Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. *MDMKDD '10*, July 2010.

[3] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou. Exploiting wikipedia as external knowledge for document clustering. *KDD '09*, July 2009.

[4] C. Li, A. Sun, and A. Datta. TSDW: Two-stage word sense disambiguation using wikipedia. *JASIST*, 64:1203–1223, June 2013.

[5] W. Meng, L. Lanfen, W. Jing, Y. Penghua, L. Jiaolong, and X. Fei. Improving short text classification using public search engines. *IUKM '13*, 8032:157–166, July 2013.

[6] M. Sahami and T. D. Heilman. A web-based kernel function for measuring the similarity of short texts snippets. *WWW '06*, May 2006.

[7] R. statistics. Post-hoc analysis for Friedman's test (R Code). http://www.r-statistics.com/2010/02/post-hoc-analysis-for-friedmans-test-r-code/, 2010. [Online; visited Apr-2014].

[8] J. Tang, X. Wang, H. Gao, X. Hu, and H. Lui. Enriching short text representation in microblog for clustering. *Frontiers of Computer Science*, 6:88–101, February 2012.

[9] TREC. Official website. http://trec.nist.gov, 2014. [Online; visited Apr-2014].

[10] G. Xu, Z. Wu, G. Li, and E. Chen. Improving contextual advertising matching by using wikipedia thesaurus knowledge. *Knowledge and Information Systems*, April 2014.

[11] M. Yazdani and A. Popescu-Belis. Computing text semantic relatedness using the contents and links of a hypertext encyclopedia. *Artificial Intelligence*, 194:176–202, January 2013.