First steps beyond the bag-of-words representation of short texts

Paolo Ferragina and Ugo Scaiella

Dipartimento di Informatica
University of Pisa, Italy
{ferragina,scaiella}@di.unipi.it

Abstract. We address the problem of enhancing the classical *bag-of-words* representation of texts by designing and engineering TAGME, the first system that performs an accurate and on-the-fly semantic annotation of short texts via Wikipedia as knowledge base. Several experiments show that TAGME outperforms state-of-the-art algorithms when they are adapted to work on short texts and it results fast and competitive on long ones. This leads us to argue favorably about TAGME's application to clustering, classification and retrieval systems on challenging scenarios like web-snippets, tweets, news, ads, etc..

1 Motivation and background

The typical IR-approach to indexing, clustering, classification and retrieval, just to name a few, is that based on the bag-of-words paradigm. In recent years a good deal of work attempted to go beyond this paradigm with the goal of improving the search experience on (unstructured) textual data. In our work we are concerned with the task of adding structure to unstructured data, consisting of the identification of sequences of terms (aka *spots*) in the input text and their annotation with a Wikipedia page. This annotation process provides a stunning contextualization of the input text so that each subsequent IR-task could be improved by leveraging the huge semantic network provided by Wikipedia. Recently several works (see e.g. [4, 6] and refs therein) addressed the problem of annotating texts with hyper-links to Wikipedia pages.

We add to this flow of work the specialty that the input texts to be annotated are *short*, namely, they are composed of few tens of terms. The context of use we have in mind is the annotation of either the snippets of search-engine results, or the tweets of a Twitter channel, or the items of a news feed, or the posts of a blog, or the advertisement messages, etc.. It is easy to argue that these poorly composed texts pose new challenges in terms of efficiency and effectiveness of the annotation process, which (1) should be very fast, because in those contexts data may be retrieved at query time and thus cannot be pre-processed, and (2) should be designed properly, because the input texts are so short that it is difficult to mine significant statistics that are rather available when texts are long. To address these issues, we have designed and implemented TAGME the first software system that, on-the-fly and with high precision/recall, annotates short texts with pertinent hyper-links to Wikipedia pages.

As an example, let us consider the following news: "Diego Maradona won against Mexico". Our goal is to detect "Diego Maradona" and "Mexico" as spots, and then hyper-link them with the Wikipedia pages which deal with the ex Argentina's coach and the football team of Mexico. TAGME uses as spots (to be annotated) the sequences of terms composing the anchor texts which occur in the Wikipedia pages, and it uses as possible senses for each spot the (possibly many) pages pointed in Wikipedia by that spot/anchor. TAGME selects among the potentially many available mappings (spot-to-page)[1] the most pertinent ones by finding a collective agreement among them via *new* scoring functions which are *fast* to be computed and *accurate* in the finally produced annotation.

A preliminary description of TAGME has been published as poster in Procs ACM CIKM 2010. TAGME is available for test at `http://tagme.di.unipi.it`. What follows is a sketch of the main ideas and experimental results concerning with TAGME, the interested reader is invited to read [3] for details.

2 The anatomy of TAGME

The annotation process of TAGME is composed by two main phases: *Anchor disambiguation* and *Anchor pruning.*

Anchor disambiguation. This is the task that judiciously cross-references each anchor $a \in \mathcal{A}_T$ found in the input text T with one pertinent page p_a of Wikipedia. TAGME selects the best association $a \mapsto p_a$ by computing a score for each possible page p_a linked to a in Wikipedia (we call $Pg(a)$ this set) that is based on a new notion of "collective agreement" between the page p_a and the pages that can be associated to all other anchors detected in T, i.e. the anchors in $\mathcal{A}_T$. This agreement is evaluated by means of a *voting scheme* that computes for each other anchor $b \in \mathcal{A}_T \setminus \{a\}$ its *vote* to the annotation $a \mapsto p_a$. Given that b may be linked to many pages in Wikipedia (i.e. $|Pg(b)| > 1$) we compute this vote as the *average relatedness* between each page p_b, potentially linked to b, and the sense p_a we wish to associate to a. However we argue that not all possible pages of b have the same (statistical) significance, so we weight each relatedness with the *commonness* of the page p_b with respect to b (denoted as $\Pr(p_b|b)$ and computed as the prior probability that b points to p_b over all links of b in Wikipedia). Hence the voting given by anchor b to the annotation $a \mapsto p_a$ is $v_b(p_a) = \frac{1}{|Pg(b)|} \cdot \sum_{p_b \in Pg(b)} rel(p_b, p_a) \cdot \Pr(p_b|b)$ where the relatedness $rel(p_b, p_a)$ between the two Wikipedia pages p_a and p_b is computed as suggested in [5] by exploiting the intersection over the incoming links to p_a and p_b. The overall score for the annotation $a \mapsto p_a$ is computed as the sum of the votes given by all anchors b in T. This score is not enough to obtain an accurate disambiguation, so we first filter out candidate pages in $Pg(a)$, via a properly set threshold, and then select the best page by deploying the commonness scores.

Anchor pruning. This step detects and possibly *prunes* some of the candidate annotations produced by the Disambiguation phase, if they are considered to be not meaningful. These "bad annotations" are detected via a simple, yet effective, scoring function that takes into account only two features: (1) the link

[1] "Diego Maradona" is the name of two persons and "Mexico" points to 154 different pages in Wikipedia.

probability $lp(a)$ of the anchor a, computed as the number of times that a is used as anchor divided by the occurrences of a in Wikipedia (as anchor or not) and (2) the *coherence* between its candidate annotation $a \mapsto p_a$ (assigned by the Disambiguation Phase) and the candidate annotations of the other anchors in T, computed with the relatedness function presented above. This *pruning score*, say $\rho(a \mapsto p)$, is then compared against a properly set threshold ρ_{NA}, so that if $\rho(a \mapsto p) < \rho_{\text{NA}}$ then that annotation for a is discarded. The parameter ρ_{NA} allows to balance recall vs precision of the annotation process and we deeply investigated its impact in our experiments.

3 Experimental evaluation

We evaluated TAGME over a set of short texts randomly drawn from Wikipedia, composed by about 20 terms (like web-snippets), and containing an average of about 10 spots. We compared the annotation produced by TAGME against the links attached by Wikipedia editors. We also evaluated TAGME on the dataset proposed in [4] that is composed by manually annotated long texts drawn from web. In annotating long texts containing more than 10 spots, TAGME processes the long input text by shifting a window of about 10 spots over it, and applying our algorithms on each window in an incremental way, so that we didn't change TAGME's architecture and TAGME is able to scale linearly with the number of anchors in the input text. For lack of space we cannot report detailed figures about the real performance of TAGME, as well as we cannot detail the comparisons against Milne&Witten's and Chakrabarti's systems[2], however here we briefly state that:

- on short texts, our annotator outperforms M&W's one by yielding an overall F-measure of about 78% with an absolute improvement of more than 8%;
- on long texts, TAGME resulted competitive (if not superior) with respect to M&W's and Chakrabarti's systems even if our "shift-based" approach clearly gives advantages to our competitors that deploy the full input text;
- a deep evaluation on the scalability and time efficiency of TAGME showed that it is able to annotate a short text of about 10 spots in 18ms, while M&W's and Chakrabarti's systems take about 95 ms and > 2 sec, respectively, on comparable hardware.

4 Applications and future works

We believe that TAGME has implications which go far beyond the enrichment of a text with explanatory links. The most interesting benefit of this annotation process is the structured knowledge attached to textual fragments that let us to leverage the semantic network provided by Wikipedia to improve IR-tasks which nowadays are mainly addressed with the bag-of-words paradigm, with all its well-known limitations.

We are currently investigating the impact of TAGME's annotation onto several application domains:

[2] A deeper evaluation is presented in our technical report [3].

- The on-the-fly labeled clustering of search-engine results (see also CLUSTY.COM, CARROT2 and the survey [2]). Most of those softwares are based on syntactic and statistical features, so they could benefit from TAGME's annotation to improve the effectiveness of the labeling and the clustering phases. For this application we are considering the deployment of spectral clustering techniques [7].
- IR systems for vertical domains. Since in this context other structured information (coming from other sources like databases, ontologies, etc.) could be available, we are investigating how to deploy such information in the TAGME-annotation process in order to further improve its performance. As a case study, we are considering the application of TAGME to the annotation of Italian medical prescriptions.
- Web Advertising. The explanatory links and the structured knowledge attached to plain-texts by TAGME could allow the efficient and effective resolution of ambiguity and polysemy issues which often occur when advertiser's keywords are matched against the content of Web pages offering display-ads.

Finally, we are studying the impact of other tools and information sources in TAGME to better relate and/or assign pages/senses to text anchors (e.g. Natural Language Processing tools, other by-products of Wikipedia— such as `DBpedia.org` or `YAGO`— or the huge amount of structured informations provided by the W3C SWEO Linking Open Data project [1]). As well as, we are currently setting up a much larger user-study over Mechanical Turk with the twofold goal of creating a manually-annotated dataset which is much larger than the one offered by [4], and of supporting the applicability of TAGME on short web-texts.

References

1. C. Bizer. The Emerging Web of Linked Data. *IEEE Int. Sys.*, 24(5):87–92, 2009.
2. C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41(3):1–38, 2009.
3. P. Ferragina and U. Scaiella. TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). In *Proc. ACM CIKM*, 1625–1628, 2010. A detailed technical report is available at `http://arxiv.org/abs/1006.3498`.
4. S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of Wikipedia entities in web text. In *Proc. ACM KDD*, 457–466, 2009.
5. O. Medelyan, D. Milne, C. Legg, and I. H. Witten. Mining meaning from Wikipedia. *Int. J. Hum.-Comput. Stud.*, 67(9):716–754, 2009.
6. D. Milne and I. H. Witten. Learning to link with Wikipedia. In *Proc. ACM CIKM*, 509–518, 2008.
7. U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.