Entity Linking on Philosophical Documents

Diego Ceccarelli[1], Alberto De Francesco[1,2], Raffaele Perego[1], Marco Segala[3], Nicola Tonellotto[1], and Salvatore Trani[1,4]

[1] Istituto di Scienza e Tecnologie dell'Informazione - CNR, Pisa, Italy
{firstname.lastname}@isti.cnr.it
[2] Istituto IMT Alti Studi di Lucca, Lucca, Italy
alberto.defrancesco@imtlucca.it
[3] Dipartimento di Scienze Umane, Università dell'Aquila, Italy
marco.segala@univaq.it
[4] Dipartimento di Informatica, Università di Pisa, Italy
trani@di.unipi.it

Abstract. Entity Linking consists in automatically enriching a document by detecting the text fragments mentioning a given entity in an external knowledge base, e.g., Wikipedia. This problem is a hot research topic due to its impact in several text-understanding related tasks. However, its application to some specific, restricted topic domains has not received much attention.
In this work we study how we can improve entity linking performance by exploiting a domain-oriented knowledge base, obtained by filtering out from Wikipedia the entities that are not relevant for the target domain. We focus on the philosophical domain, and we experiment a combination of three different entity filtering approaches: one based on the "*Philosophy*" category of Wikipedia, and two based on similarity metrics between philosophical documents and the textual description of the entities in the knowledge base, namely cosine similarity and Kullback-Leibler divergence. We apply traditional entity linking strategies to the domain-oriented knowledge base obtained with these filtering techniques. Finally, we use the resulting enriched documents to conduct a preliminary user study with an expert in the area.

Keywords: Entity Linking, Entity Filtering, Information Search and Retrieval, Document Enriching

1 Introduction

In the latest years, document enriching via *Entity Linking (EL)* has gained increasing interests due to its impact in several text-understanding related tasks, *e.g.,* web search, document classification, etc [11, 12]. The EL problem has been introduced in 2007 by Mihalcea and Csomai [8] and consists in linking short fragments of text within a document to an entity listed in a given Knowledge Base (KB). The authors also propose to consider each Wikipedia article as an entity, and the title or the anchor text of the hyperlinks pointing to the article as potential mentions to the entity.

In **Dresden**→ [http://en.wikipedia.org/wiki/Dresden], **Schopenhauer**→ [http://en.wikipedia.org/wiki/Arthur_Schopenhauer] became acquainted with the **philosopher**→ [http://en.wikipedia.org/wiki/Philosopher] and **freemason**→ [http://en.wikipedia.org/wiki/Freemasonry], **Karl Christian Friedrich Krause**→ [http://en.wikipedia.org/wiki/Karl_Christian_Friedrich_Krause]

Fig. 1: Example of annotated document

A typical EL system works in three steps: *i)* **Spotting**: the document is processed in order to detect a set of potential mentions (also referred as *surface forms* or *spots*), and for each mention a list of candidate entities is produced; *ii)* **Disambiguation**: for each potential mention with more than one candidate entity, a single entity is selected. This is done by trying to maximize the coherence among the selected entities; *iii)* **Filtering**: only the most relevant annotations, i.e., the mentions linked with some entity, are selected, filtering out the irrelevant ones by using some measure of annotation confidence/importance. Due to the ambiguity of natural languages, the EL task is not trivial. In fact the same mention could refer to more than one entity (polysemy) and the same entity could be referred by more than one mention (synonymy).

Let us introduce an example to describe how the EL process works. Figure 1 shows a semantically enriched document produced by an EL system. In the reported text there are mentions (e.g., **Dresden**, **Schopenhauer**, or **freemason**) linked to their semantic concept by using the URI or the identifier in the KB, in our case Wikipedia. For example, the spot **Dresden** is linked to `http://en.wikipedia.org/wiki/Dresden`. It is worth noting that the mention **Dresden** could refer to many other meanings, as we can see by looking at the corresponding Wikipedia disambiguation page[5].

Now we introduce some notations used thereinafter in the paper. A Knowledge Base is a collection of entities, where each entity represents an artifact or a concept in the real world. An entity e is described by the following attributes:

- a **Uniform Resource Identifier**: univocally identify the entity in KB (e.g., the url "`http://en.wikipedia.org/wiki/Dresden`" identify the entity Dresden);
- a **description**: text describing what the entity represents, usually the content of its Wikipedia page (e.g., "Dresden is the capital city of...");
- a set of **related entities** that are connected to the given entity, usually derived from Wikipedia links (e.g., Germany is linked by Dresden);
- a set of **surface forms**, the fragments of text used to refer the entity (e.g., "A. Schopenhauer" and "Arthur Schopenhauer" are both surface forms for `http://en.wikipedia.org/wiki/Arthur_Schopenhauer`);
- a set of **categories**, organized in a taxonomy, the entity belongs to (e.g., Schopenhauer belongs to the categories "Idealists" and "German atheists").

The entity linking task consists in finding an annotation function f_{EL} that, given KB and a raw text document d, returns an enriched version of the document d_e which includes also a list of annotations. Each annotation is described by a tuple $< start, end, text, entity >$, where:

[5] `http://en.wikipedia.org/wiki/Dresden_(disambiguation)`

- **start** is the starting offset of the annotation in the document;
- **end** is the ending offset of the annotation in the document;
- **text** is the surface form of the entity detected in the document;
- **entity** is the URI of the entity detected in the document.

The research question behind this paper is the following: let us assume to have a collection of documents about a particular topic to enrich, e.g., Philosophy. Could we exploit the knowledge about the topic to improve the effectiveness of the entity linking process? The solution we propose works a priori on the Knowledge Base (KB) used for generating the EL model. The idea is to consider only the entities relevant for a target topic t of the documents we are going to annotate. These entities form a new domain-specific Knowledge Base, that in the following we refer to **Topical Knowledge Base**.

To the best of our knowledge, we are the first to investigate how to perform topical EL by prefiltering a general knowledge base. Mirylenka and Passerini [10], and later Miao *et al.* [7], applied EL techniques on the domain of scientific publications: in [10] authors propose a method of organizing the search results into concise and informative topic hierarchies. They obtain the hierarchies by annotating the entities in a document with Wikipedia Miner [9] – an open source entity linking tool. STICS [4] is a system that enriches news with entities and uses them for improve the browsing and provide entity analytics of what is happening in the world. Finally, Ernst *et al.* [2] applied entity linking on health and life sciences, through the KnowLife portal, a large KB automatically constructed from Web sources.

2 The Knowledge Base Topic-Filtering Problem

Let $f_{EL}(KB, d)$ be an annotation function that, given a Knowledge Base KB and a document d, produces an enriched version of the document d_e, and let $\sigma(f_{EL}(KB, d))$ be an effectiveness measure of the annotation function, i.e. a common information retrieval quality metrics such as precision.

Given a collection of documents D_t related to a topic t, our objective is to find a subset KB_t of the knowledge base KB, such that:

$$\forall d \in D_t,\ \sigma(f_{EL}(KB_t, d)) \geq \sigma(f_{EL}(KB, d))$$

$$|KB_t| \ll |KB|$$

The topical knowledge base KB_t is obtained by filtering KB through a function $\phi(KB, t)$. Since KB is a collection of entities $\{e_1, e_2, \cdots, e_n\}$ and ϕ filter each entity independently from the others, we can thus define:

$$KB_t = \bigcup_{e \in KB} \phi(e, t)$$

Our claim is that such a function ϕ can improve the effectiveness of the entity linking task for the topic t. In particular, we propose three filtering methods:

Cosine Similarity Filter, *Kullback-Leibler Divergence Similarity Filter*, and *Category Filter*. The first two approaches exploit the textual similarity[6] between the documents in D_t and the description of the entity in KB (averaging the result with respect to the collection D_t). The latter exploits the Wikipedia Category Graph in order to detect how far are the categories the entity belongs to and the root category of the topic being considered.

Cosine Similarity Filter

The cosine similarity filter measures the similarity between two vectors. Let d be a document belonging to a collection of documents D related to a topic t, e_{desc} be the textual description of an entity (e.g., the text in its Wikipedia page), $w_{k_i}^{(d)}$ the weight associated with a term-document pair (k_i, d), $w_{k_i}^{(e_{desc})}$ the weight associated with a term-entity pair (k_i, e_{desc}). Then, in the textual similarity context, the cosine similarity is defined as:

$$sim(d, e_{desc}) = \frac{\sum\limits_{k_i \epsilon V} w_{k_i}^{(d)}}{\sqrt{\sum\limits_{k_i \epsilon V} {w^2}_{k_i}^{(d)}}} \times \frac{\sum\limits_{k_i \epsilon V} w_{k_i}^{(e_{desc})}}{\sqrt{\sum\limits_{k_i \epsilon V} {w^2}_{k_i}^{(e_{desc})}}} \tag{1}$$

where $V = \{k_1, \ldots, k_n\}$ is the vocabulary of the terms, n is the number of distinct terms in the document collection and k_i be a generic term. The weights $w_{k_i}^{(d)}$ and $w_{k_i}^{(e_{desc})}$ are computed with the *tf-idf* [6] formula as in the following, using the inverse document frequency of the term in Wikipedia (idf_w)

$$w_{k_i}^{(d)} = tf^{(d)}(k_i) \times idf_w(k_i) \tag{2}$$

$$w_{k_i}^{(e_{desc})} = tf^{(e_{desc})}(k_i) \times idf_w(k_i) \tag{3}$$

The cosine similarity ranges in $[0, 1]$, with the maximum similarity reached at 1.

Kullback-Leibler Divergence Filter

The Kullback-Leibler Divergence (KLD) Filter measures the relative entropy of two different probability distributions associated to the same event space. Let d and e_{desc} be defined as in *Cosine Similarity Filter*, $P_{k_i}^{(d)}$ be the probability of a term k_i in a document, and $P_{k_i}^{(e_{desc})}$ be the probability of a term k_i in an entity description, the Kullback-Leibler divergence is formulated in [5] as follows:

$$\sum_{k_i \epsilon V} \left\{ P_{k_i}^{(e_{desc})} \times \log \frac{P_{k_i}^{(e_{desc})}}{P_{k_i}^{(d)}} \right\} \tag{4}$$

where V is the vocabulary $V = \{k_1, \ldots, k_n\}$ representing the set of all distinct index terms in the collection of documents, and the $P_{k_i}^{(d)}$ and $P_{k_i}^{(e_{desc})}$ probabilities

[6] In Information Retrieval, the text similarity between document-query pair, is a score aiming to provide a degree of similarity of a document with respect to an user information need.

respectively defined as in the following:

$$P_{k_i}^{(d)} = \frac{w_{k_i}^{(d)}}{\sum_{k_i \epsilon V} w_{k_i}^{(d)}} \tag{5}$$

$$P_{k_i}^{(e_{desc})} = \frac{w_{k_i}^{(e_{desc})}}{\sum_{k_i \epsilon V} w_{k_i}^{(e_{desc})}} \tag{6}$$

where $w_{k_i}^{(d)}$ and $w_{k_i}^{(e_{desc})}$ are respectively computed as in Equations (2,3). For the sake of simplicity in this paper we are using the original formulation of the KLD, which is not symmetric (i.e., $KLD(d, e_{desc}) \neq KLD(e_{desc}, d)$). A more reliable implementation could be the symmetrised or the Jensen-Shannon divergence because they consider also the similarity between the textual description of the entity and the document.

Category Filter

The Category Filter takes advantages of the Wikipedia category graph and of the list of categories each entity belongs to. This information is used to compute the shortest path (and so the minimum distance) of an entity from a set of highly relevant category node (the root of the visit) for the topic being considered (*e.g., Philosophy*[7]). Each Wikipedia article can appear in more than one category, and each category can appear in more than one parent category. Multiple categorization schemes co-exist simultaneously. In other words, categories do not form a strict hierarchy or tree structure, but a more general directed acyclic graph (DAG).

In particular, given $G = (C, E)$ be the Wikipedia category graph with C the category nodes and E the direct connection between the categories, let us define $C^{(t)} = \{c_1^{(t)}, c_2^{(t)}, \ldots, c_m^{(t)}\}$ be the set of highly relevant categories relative to the topic t, with $C^{(t)} \subset C$. The minimum distance of each wikipedia entity from the categories in $C^{(t)}$ can so be computed by exploiting a breadth-first search (BFS) visit of the graph G, starting from the nodes in $C^{(t)}$. Let us define such a method with the function φ.

$$\varphi(c_i^{(t)}) = \{BFS(G, C^{(t)})\} \quad \forall i \in 1, \ldots, n \tag{7}$$

where n is equal to $|C|$.

3 Experiments

In the following we introduce the philosophical document adopted as a reference document for the topic $t = Philosophy$. This document is used both for filtering

[7] http://en.wikipedia.org/wiki/Category:Philosophy

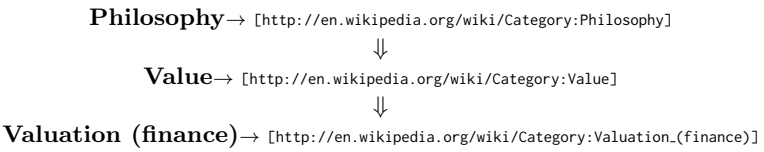


Fig. 2: Odd path in the category graph of wikipedia

by textual similarity and for performing the user study described in the Section 4. Then we describe the methodology adopted to build the filtered KB and the differences in the *EL* annotations obtained by using the traditional KB and the filtered one.

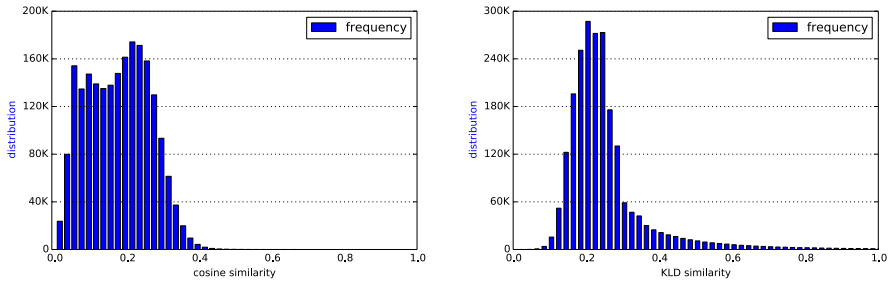
3.1 Reference document

We adopt a philosophical text written by the philosopher *Ludwig Wittgenstein* during the middle of the last century as the reference document for the topic t being considered. The title of the book is *On Certainty* and it is a collection of aphorisms discussing the relation between knowledge and certainty. The book is composed by 676 paragraphs, with an average of 243 characters and 46 terms per paragraph. Since each paragraph is long enough and contains several philosophical notions, we consider it as an independent document d of D_t.

3.2 Filtering methods

In order to evaluate the impact of the proposed filtering on the KB and to gain some insights on the thresholds to adopt, we perform a study to measure the frequencies of the entities that pass each filtering strategy in isolation, by considering different values for the thresholds. Given the problem formalization described in Section 2, we apply each filtering method to each entity in KB, computing a score that expresses how *far/similar* is the entity from the topic t. We compute the textual similarity by applying the *Cosine Similarity* and the *Kullback-Leibler Distance* between the reference document described above and the description of the entity in KB, *i.e.,* the content of the Wikipedia article. The *Category Filter* computes the minimum depth of all the categories each entity belongs to. The category graph as well as the categories related to an entity are taken from the KB.

Figure 3 reports the application of the *Cosine Similarity* (Figure 3a) and *Kullback-Leibler Divergence* (Figure 3b) filters to the KB. The former obtains maximum similarity when the scores is 1, while the latter when the score is 0. The two figures show that the cosine similarity is more spread out along the X axis (the confidence score thresholds) than *KLD*, which is indeed very thin-tailed. Finally, Figure 4 shows the *category* filter application, with the depth distribution of the categories in the category graph given the root node *Philosophy* (Figure 4a) and the distribution of the entities according to the minimum depth



(a) Cosine similarity distribution (b) Kullback-Leibler distribution

Fig. 3: Distribution using the textual similarity filtering

of the categories each entity belongs to (Figure 4b). The outcome of this figure is quite surprising: the category graph (which is a DAG according to Wikipedia) seems to interconnect categories that are not strictly related each other. As a matter of fact, take a look at the odd path in Figure 2, where the category *Valuation (Finance)* is reached by traversing only two descendant link starting from the *Philosophy* category node. This evidence clearly explains also why so many categories ($\approx 750k$) can be reached by descendant traversing the category graph from the Philosophy node, with at most 10-11 steps.

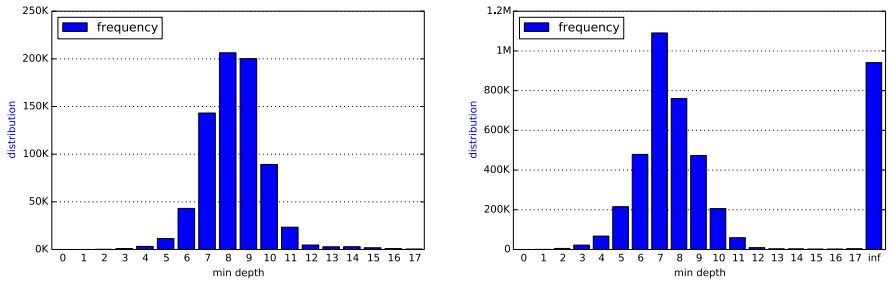
3.3 Filtering approach adopted

According to the Wikipedia Philosophical Portal [8], there are about $\approx 15k$ philosophical articles on the total of $\approx 4.35M$ articles. Our intuition is that by restricting ourself to this topic specific subset of articles may not be sufficient from an *EL* perspective. Indeed by considering also articles very related to the topic could results in an improvements of the EL effectiveness, due to a disambiguation phase which make use also of entities only marginally related with the topic t. Thus, our suggestion is to select 2 times the number of entities wikipedia says to belong to the philosophical portal (i.e., our goal is to select $\approx 30k$ entities). Obviously this approach is reasonable only because we are investigating a new, unexplorated research direction. A more general way of solving the problem would lead to not decide a-priori the number of entitites to select, but it should depends on the domain and on how the domain is covered in Wikipedia. The same reasoning would lead to choose the thresholds for the textual similarity strategies depending only from domain-specificity of the entities.

Hence, for filtering out entities not related to the topic t, we adopt the following approach:

1. We make use of the category filter to select entities belonging to the topic and to build the base KB. This choice is motivated by the high accuracy we expect from the category graph in selecting entities relevant for the topic,

[8] `http://en.wikipedia.org/wiki/Portal:Philosophy`



(a) Category distribution (b) Entity Distribution

Fig. 4: Distributions using the category filtering

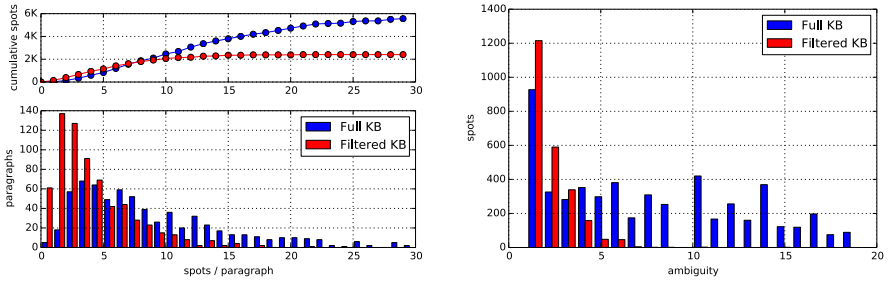
because the categories are manually assigned (and validated) by the user to the entities. In order to maximize the probability of selecting only topic related entities, we exploit a very aggressive threshold value of 3 (*i.e.,* 3 is the minimum distance between at least one of the category the entity belongs to and the root category of the topic, which is in our case the Philosophical category). By using this threshold the filter selects $\approx 28k$ entities.

2. We expand the KB by considering also textual similarities with the reference document described in Section 3.1. The idea here is that some entities in wikipedia could be misclassified (i.e., their categories could not reflect the real topic of the entity or the entity could miss of some relevant categories). In order to apply such an expansion, we adopt the two textual filtering approaches described in Section 3.3 by combining them together, *i.e.,* only entities that pass both the filters are added to the base KB. Since our target is to select $\approx 30k$ entities, and we have $28k$ from the Categories, we add the missing $2k$ considering the intersection between the subset of entities with the highest cosine and similarity and the subset of entities with the highest KLD similarity. We select the subsets finding two thresholds so that the sets have approximately the same size. We then filter the cosine similarity (respectively the Kullback-Leibler divergence) with a threshold of 0.35 (0.125) which let $\approx 52k$ ($\approx 79k$) entities pass the filter.

The resulting KB is made up of $\approx 30k$ entities, primarily selected by investigating the category graph and expanded with highly similar (from a language point of view) side entities. The size reduction compared to the full KB is $\approx 99\%$.

3.4 Entity Linking differences

Traditional entity linking strategies are applied to the reference document in order to evaluate how the annotation process is affected by the domain-oriented knowledge base obtained with the proposed filtering approach. We used the EL Dexter framework [1] to annotate the philosophical document with both the traditional KB and by plugging into it the domain oriented KB. Each paragraph was annotated independently from the others, so that we are able to investigate



(a) Spots distribution (b) Ambiguity distribution

Fig. 5: EL annotation differences using the full KB and the filtered one

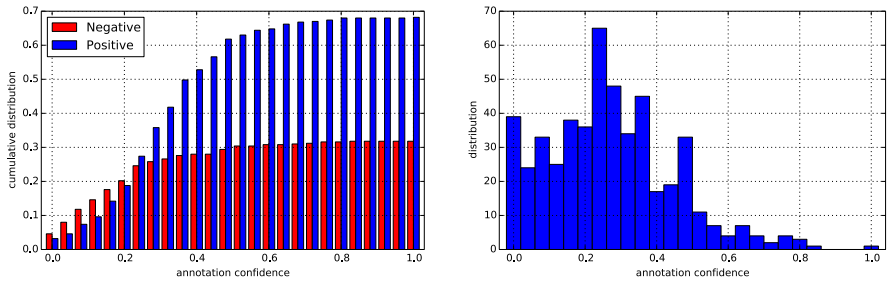
the difference in the annotation process by looking at two important factors: how many spots per paragraph the system is able to annotate and how much each spot is ambiguous before performing the disambiguation phase, *i.e.,* how many entities on average are candidated for each spot. In Figure 7 we report these factors by comparing the two KB solutions. As expected, Figure 5a shows that the distributions of spots per paragraph is very different: on average by using the filtered KB, the EL system annotates less spots per paragraph than using the full KB, and this evidence is very clear if we look at the number of cumulative spots annotated. Indeed by using the full KB, the EL system annotates approximately 3 times more spots than using the filtered one. If we look at the distributions of the ambiguity per spot, another important aspect arises: on average the EL system which uses the filtered KB selects far less entities as spots, with a 50% probability to select only one candidate per spot and 25% probability to select two candidates for a spot. The latter evidence is really important because the disambiguation phase is simpler when the ambiguity is low (and it is worth to notice that 50% of the spots do not need a disambiguation at all due to a single candidate per spot selected).

4 User study

For assessing the quality of the linking performed using the filtered KB, we set up a user study experiment. We selected the first 110 paragraphs from the *On Certainty* book, and we annotated each paragraph using a model generated from the filtered KB. Each paragraph was annotated using a dictionary generated from the Wikipedia anchors, and, in case of ambiguous spots, we disambiguated using the TAGME disambiguation algorithm [3].

On average we annotated 4.54 entities per paragraph (500 in total). We designed a simple web application that allows a user to evaluate the annotations. The application allows to browse the paragraphs in two different ways:

Document based browsing the user can visualize a paragraph and move to the previous, or the next. Annotated spots are highlighted and if the user clicks on a spot, a description of the annotated entity pops out;



(a) Assessment distribution by confidence (b) Annotation confidence distribution

Fig. 6: EL assessment distributions using filtered KB

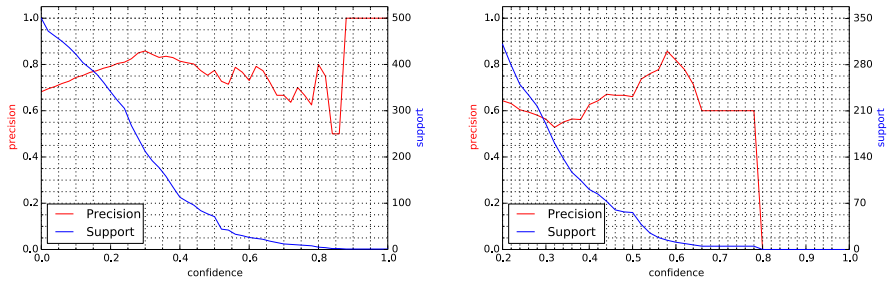
Entity based browsing If the user clicks on the name of the entity, then an entity based view is presented: this page presents a description of the entity, and then a list of paragraphs where the entity is mentioned. For each paragraph the user can visualize the spot that was annotated with the entity, and the text around the annotation.

The user can mark an annotation as *good* or *bad*, simply by clicking on it. One click means *good* and the annotation is highlighted in green, while an additional click means *bad* and the annotation is highlighted in red.

We asked an expert human annotator in the field to judge the annotations. Subsequently we performed the same evaluation, but the EL was performed using the full KB. We obtained on average 9.72 annotations per paragraph (1070 in total). This high number is due to the fact that we did not pre-filter the annotations in any way. To speed up the expert assessment, we decided to remove the annotations with a confidence (i.e., a score assigned by the EL system which express the certainty of the annotation) lower than 20%. This threshold is absolutely reasonable since usually TAGME discards annotations with a confidence lower than 50%. The number of annotations per document decreased from 9.72 to 2.82 (globally from 1070 to 311). Moreover, we automatically copied the assessments relative to the same annotations (*i.e.,* spots occurring in the same place of the text linked to the same entity) from the previous judgment performed by the expert annotator. This avoided the evaluation of 113 (36%) annotations.

Figure 6b shows the distribution of the annotations by their confidence score. We can observe that a traditional EL system encounters some problem on working with a filtered KB. Indeed the annotation confidence is on average quite low, thus suggesting that the mutual reinforcement of the entities in the disambiguation phase has still a lot of space for improvement when working on a topic based KB. It is worth to notice that the disambiguation strategy adopted (TAGME) would have discarded the majority of the annotations by applying its threshold value (50%) on the annotation confidence.

Figure 6a on the other hand illustrates the distribution of the two assessment classes by the confidence score of the relative annotations. We can identify a clear correlation between the positive class and the confidence score (higher is better).



(a) Filtered KB (b) Full KB

Fig. 7: EL annotation effectiveness

This can be explained by the fact that high confidence scores are assigned to entities strongly related with other entities in the annotated document, thus resulting in a more precise annotation which is less prone to errors.

Finally, Figure 7 depicts the annotation effectiveness of the two EL systems, the first using the full KB and the second using the filtered one. The effectiveness is evaluated at different values of confidence in order to study the best threshold to adopt for filtering out bad annotations and maximizing the effectiveness of the annotation process. In the figures we show the *support*, *i.e.,* the number of assessments with a confidence higher or equal to the threshold adopted, and the *precision*, *i.e.,* the fraction of positive assessments over the sum of positive and negative assessments. Figure 7a clearly depicts that the precision starts from a value of ≈ 0.7 (obtained without using any threshold on the confidence, thus having the maximum support) to a maximum of ≈ 0.85 (using a threshold of 0.3, corresponding to a support of ≈ 430). Further values slowly decrease in precision, but – more importantly – decrease in support. A very low support means very few annotations, and we should avoid such compromise.

Figure 7b studies the behavior of the EL system that makes use of the full KB. Here the precision ranges from a value of ≈ 0.65, starting from a confidence value of 0.2 (remind that annotations with a confidence below this threshold were discarded) up to a maximum of ≈ 0.85 obtained with a confidence value of 0.57 (with a support of 12). The latter results clearly depict how using only precision is not enough to measure the effectiveness of a system: in fact, the support value of 15 means that only 5% of the annotations are considered, resulting in a very poor document enriching (*i.e.,* 0.14 average annotations per paragraph). By comparing the behavior of the two systems it is evident that the strategy to build a topical KB is clearly a key idea for performing EL on a set of topic specific documents. This is supported by the fact that we obtained a consistent improvement in terms of precision without penalizing too much the support.

5 Conclusions

In this paper we study how to apply entity linking on a collection of documents about a particular topic. Our thesis is that pre-filtering a general knowledge base

keeping only the entities that are relevant for the topic, and then building the entity linking model only from these entities could improve the annotation performance. We propose three strategies for filtering the knowledge base, two based on textual similarity between the topic and the entities (i.e., cosine similarity and KL-divergence) and one based on the Wikipedia categories (i.e., considering only the categories that belong to the selected topic). We perform some preliminary experiments on the topic Philosophy, combining the three methods and performing the linking on the resulting filtered knowledge base. Finally, in a user-study performed by an expert in the area, we compare the annotation performance on a philosophic document collection using a traditional knowledge base and the one filtered by our approach. The results confirm that the proposed technique is a promising idea for performing EL on a set of topic specific documents.

References

1. D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani. Dexter: an open source framework for entity linking. In *Proceedings of the Sixth International Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR)*, 2013.
2. P. Ernst, C. Meng, A. Siu, and G. Weikum. Knowlife: A knowledge graph for health and life sciences. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 1254–1257. IEEE, 2014.
3. P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of CIKM*, 2010.
4. J. Hoffart, D. Milchevski, and G. Weikum. Stics: searching with strings, things, and cats. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1247–1248. ACM, 2014.
5. S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.
6. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2009.
7. Q. Miao, Y. Meng, L. Fang, F. Nishino, and N. Igata. Link scientific publications using linked data. In *Semantic Computing (ICSC), 2015 IEEE International Conference on*, pages 268–271. IEEE, 2015.
8. R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.
9. D. Milne and I. H. Witten. An open-source toolkit for mining wikipedia. *Artificial Intelligence*, 194:222–239, 2013.
10. D. Mirylenka and A. Passerini. Navigating the topical structure of academic search results via the wikipedia category network. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 891–896. ACM, 2013.
11. P. Pantel and A. Fuxman. Jigs and lures: Associating web queries with structured entities. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.
12. G. Weikum and M. Theobald. From information to knowledge: harvesting entities and relationships from web sources. In *Proceedings of PODS*, 2010.