SEL: a Unified Algorithm for Entity Linking and Saliency Detection

Salvatore Trani
ISTI–CNR, Pisa, Italy
s.trani@isti.cnr.it

Diego Ceccarelli
Bloomberg LP
dceccarelli4@bloomberg.net

Claudio Lucchese
ISTI–CNR, Pisa, Italy
c.lucchese@isti.cnr.it

Salvatore Orlando
Università Ca' Foscari
Venezia, Italy
orlando@unive.it

Raffaele Perego
ISTI–CNR, Pisa, Italy
r.perego@isti.cnr.it

ABSTRACT

The *Entity Linking* task consists in automatically identifying and linking the entities mentioned in a text to their URIs in a given Knowledge Base, e.g., Wikipedia. Entity Linking has a large impact in several text analysis and information retrieval related tasks. This task is very challenging due to natural language ambiguity. However, not all the entities mentioned in a document have the same relevance and utility in understanding the topics being discussed. Thus, the related problem of identifying the most relevant entities present in a document, also known as *Salient Entities*, is attracting increasing interest.

In this paper we propose *SEL*, a novel supervised two-step algorithm comprehensively addressing both entity linking and saliency detection. The first step is based on a classifier aimed at identifying a set of candidate entities that are likely to be mentioned in the document, thus maximizing the precision of the method without hindering its recall. The second step is still based on machine learning, and aims at choosing from the previous set the entities that actually occur in the document. Indeed, we tested two different versions of the second step, one aimed at solving only the entity linking task, and the other that, besides detecting linked entities, also scores them according to their saliency. Experiments conducted on two different datasets show that the proposed algorithm outperforms state-of-the-art competitors, and is able to detect salient entities with high accuracy.

Keywords

Entity Linking; Salient Entities; Machine Learning

1. INTRODUCTION

Lately, much research has been spent to devise effective solutions to Entity Linking (EL). The task, also known as *Wikification*, has been introduced by Mihalcea and Csomai [9], and consists in finding small fragments of text (hereinafter named *spots* or *mentions*) referring to an entity that is listed in a given knowledge base, e.g., Wikipedia. Natural language ambiguity makes this task non trivial. Indeed, the same entity may be mentioned with different text fragments, and the same mention may refer to one of several entities.

EL is strictly correlated with another task, referred to as *document aboutness* problem [11] or *Salient Entities* (SE) discovery problem [6], which goal is labeling the entities mentioned in the document according to a notion of saliency, where the most relevant entities are those that have the highest utility in understanding the topics discussed.

As an example, consider the annotations performed by an EL algorithm that uses Wikipedia on the following text:

> **Maradona** (→Diego_Maradona) played his first **World Cup tournament** (→FIFA_World_Cup) in 1982, when **Argentina** (→Argentina_national_football_team) played **Belgium** (→Belgium_national_football_team) in the opening game of the **1982 Cup** (→1982_FIFA_World_Cup) in **Barcelona** (→Barcelona).

Such an algorithm performs the EL task by first spotting the fragments of text that are likely to refer to some entity, e.g., spots **Maradona** or **Belgium**. Indeed, in this phase multiple candidate entities can be generated for each spot. Then, the algorithm proceeds by trying to link each spot to the correct entity, e.g., links the spot **Maradona** to the corresponding Wikipedia page[1]. Due to the presence of multiple candidates for each spot and to the inherent ambiguity of natural language, the disambiguation phase of the EL process is not trivial, e.g., the mention **Belgium** does not refer to its most common sense, i.e., the country, but rather to its national football team[2]. A final stage of pruning discards annotations that are considered not correct or consistent with the overall interpretation of the document.

As previously stated, Salient Entities (SE) discovery can be combined with EL. The easiest integration is to perform the SE discovery as a subsequent step to EL, by finally choosing the most relevant entities that have high utility in understanding the topics being discussed among the set of entities returned by the EL algorithm. However, we claim this pipeline approach is somehow limiting since the disambiguation could benefit from the saliency signal. In our example,

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

[1] https://en.wikipedia.org/wiki/Diego_Maradona
[2] https://en.wikipedia.org/wiki/Belgium_national_football_team

the most relevant entities are probably the ones referred by mentions **Maradona** and **1982 Cup**.

Entity saliency impacts on information extraction from text in a broader sense. Consider for example a semantic clustering approach where linked entities are exploited to provide a high-level summary of each document. In this application scenario the capability of weighting entities on the basis of their saliency is crucial. In addition, the knowledge about the saliency of entities recognized by an EL algorithm in a document should also impact on the evaluation of the effectiveness of the EL algorithm itself. Let us come back to the previous example where the entity **1982 Cup** provides much more information about the document than the entity **Barcelona**. Thus, an EL algorithm that links only the mention **1982 Cup** should be preferred in terms of effectiveness to another algorithm that only links the spot **Barcelona**.

In this paper we propose a novel supervised *Salient Entity Linking* (*SEL*) algorithm to comprehensively address EL and SE detection. The *SEL* algorithm entails two steps: *Candidate Pruning* and *Saliency Linking*. During the *Candidate Pruning* step, a classifier is used to prune the large set of candidate entities generated by the spotting phase. The aim is to detect a relatively small collection of candidates that encompasses all the entities actually mentioned in the document. Thus the emphasis is on training a classifier able to achieve a good precision without hindering recall. The proposed approach has proved to outperform heuristic methods that prune unlikely candidates on the basis of simple likelihood measures such as *commonness* or *link probability* [9, 10]. The *Saliency Linking* step also exploits machine learning, and, in addition to addressing EL, it is able to predict the saliency of the entities that survived the *Candidate Pruning* step. Thanks to the *Candidate Pruning* step, the candidate set processed during the *Saliency Linking* step is less noisy and smaller in size, which allows to use more complex and powerful graph-based entity correlation features.

The experiments conducted on two different datasets show that *SEL* outperforms state-of-the-art competitors in the EL task. In addition, it is able to detect salient entities with high accuracy. Since both steps of the algorithm are based on machine learning, we also analyzed in depth feature importance, and we took into consideration feature extraction costs. We show that an efficient and effective classifier for the first step can be trained on the basis of a small and easily computable set of features. This is particularly important since the classifier must be applied to a very large set of initial candidates. On the other hand, in the second step we have a reduced number of survived candidates and we benefit from the exploitation of further graph-based features, which are more expensive to compute, but which are proved to be very effective for improving the quality of entity linking and saliency detection.

In summary, the main contributions of this paper are:

- a novel *Salient Entity Linking* (*SEL*) algorithm, that accurately estimates entity saliency and outperforms state-of-the-art EL techniques by providing a comprehensive solution to the EL and the SE detection problems;
- an evaluation of a wide set of heterogeneous features, including novel features, used to represent entities within the machine learning algorithms adopted;
- a novel dataset of news manually annotated with entities and their saliency, hereinafter publicly available to the research community.

2. RELATED WORK

Entity Linking. Entity Linking algorithms usually work by following a well defined schema, that could be roughly summarized in three steps: *spotting*, *disambiguation* and *pruning*. *Spotting* detects potential mentions in a text and, for each mention, produces a list of *candidate entities*. *Disambiguation* aims at selecting a single entity for each mention produced in the previous step, by trying to maximize some *coherence* measure among the selected entities in the document. *Pruning* detects and removes non-relevant annotations in order to improve the precision of the system. In performing the three steps, EL algorithms rely on three effective signals: *(i)* the probability for a mention to be a link to an entity (*link probability*); *(ii)* the prior probability for a mention to refer to a specific entity (*commonness*); *(iii)* the *coherence* among the entities in a document, e.g., estimated by the Milne-Witten *relatedness* [10]. In addition to annotate mentions to the entities, EL algorithms usually assign to each annotation a *confidence score*, roughly estimating the correctness of the annotation.

Several EL approaches have been proposed following the problem formalization given by Mihalcea and Csomai with *Wikify* [9]. A substantial improvement has been the Wiki-Miner approach proposed by Milne and Witten [10]. It works by first identifying a set of non-ambiguous mentions and then using this set to disambiguate the ambiguous ones. Ferragina and Scaiella proposed an improved approach called Tagme [5], which tries to find a collective agreement for the best candidates using a voting scheme based on the the Milne-Witten relatedness. Candidate entities with a coherence below a given threshold are discarded, and for each mention the one with the largest commonness is selected. In Spotlight [8], Mendes *et al.* represent each entity with a context vector containing the terms from the paragraphs where the entity is mentioned; they also exploit NLP methods, removing all the spots that are only composed of verbs, adjectives, and prepositions. In Wikifier 2.0 [2] (which is an extension of [13]), Cheng and Roth use a machine learning based hybrid strategy to combine local features, such as commonness and TF-IDF between mentions and Wikipedia pages, with global coherence features based on Wikipedia links and relational inference. This system combines Wikipedia pages, gazetteers, and Wordnet. In AIDA [7], Hoffart *et al.* proposed a weighted mention-entity graph for collective disambiguation. This model combines three features into a graph model: entity popularity, textual similarity (keyphrase-based and syntax-based) as well as coherence between mapping entities. The authors also published a manually annotated dataset for EL, named AIDA-CoNLL 2003. In WAT [12] authors extended Tagme with a new spotting module (using gazetteers, named-entity recognition analysis and a binary classifier for tuning performance), voting-based and graph-based disambiguation approaches as well as a pruning pipeline. Note that neither the source code nor a remote annotation service of WAT is publicly available. One of the main conclusions from their experiments was that while many systems focused on improving disambiguation, the spotter and the pruner are actually responsible for introducing many of the false positives in the EL process. A thorough overview and analysis of the main approaches to EL and their evaluation is presented by Shen *et al.* [15].

Entity Saliency. The problem of understanding the main topics of a document has been the goal of many IR tasks, including latent semantic topics and text summarization. In this work we tackle the related task of finding the most important entities mentioned in a given document. This task has previously been referred to as *document aboutness* [6] or *salient entity discovery* [14] problem.

Gamon *et al.* [6] studied the *aboutness* problem referred to the named entities occurring in Web pages. The approach used is partially inspired by [11], where click-through data are exploited to rank named entities mentioned in queries. The authors estimate the entity saliency for a Web page by exploiting the click-through recorded in a query log. Roughly, a document is considered to be relevant for a given entity when it is returned by a Web search engine and clicked by multiple users in answer to queries mentioning the entity. A number of text-based features are proposed in the paper, most of them applicable only to a Web scenario, e.g., url depth. In such work entities are just pieces of text (and not entities listed in a given knowledge base) and the disambiguation problem is not tackled at all.

When entities in a knowledge base such as Wikipedia are considered, rich contextual information coming from its graph structure can be fruitfully exploited. Given the set of entities occurring in a document, an *entity graph* can be built by projecting the subgraph of the knowledge base graph including all the entities possibly mentioned in the document. Entities can finally be ranked according to some measure of their importance in such a graph.

Dunietz and Gillick [3] proposed a method for classifying salient entities mentioned in news by exploiting graph-based measures. They show that the eigenvector centrality computed on the mentioned entities can slightly improve the performance of a binary classifier aimed at discriminating salient entities with respect to a classifier learned with text-based features only. The same task is addressed in [14], where text-based features are fruitfully complemented with graph-based ones to improve accuracy. The work by Dunietz and Gillick is closely related to ours but, in order to automatically generate the ground truth, they consider as salient entities those mentioned in the abstract of the news. Thus, the authors cannot use features related to the position of the mention for predicting the saliency, and how the graph-based and other features contribute to improve the classification accuracy. We instead exploited a manually assessed dataset that allows us to perform this analysis. Moreover, their paper assumes to know in advance the correct entities mentioned in the document, and addresses only the problem of ranking them by saliency. Instead we addressed comprehensively the EL and SE problems, and studied the importance of different features for identifying the correct entities mentioned as well as their saliency.

3. THE SALIENT ENTITY LINKING ALGORITHM

Let $\mathcal{KB}$ be a knowledge base with a set of entities $\mathcal{E}$. The EL problem is to identify the entities $\mathcal{E}_D \subseteq \mathcal{E}$ mentioned by the *spots* S_D of a given document D. As in state-of-the-art approaches, Wikipedia is used as knowledge base and every Wikipedia article is considered as an entity. Entities that are not in Wikipedia are not linked (i.e., we do not take into account the NIL problem).

In this paper the *saliency* $\sigma(e|D)$ of the entities e mentioned in a document D is also considered. Without loss of generality, we define the domain of function σ as the set $\{0, 1, 2, 3\}$, with the following meaning:

- **3 - Top Relevant**: the entity describes the main topics or the leading characters of a document;
- **2 - Highly Relevant**: these are satellite entities that are not necessary for understanding the document, but they provide important facets;
- **1 - Partially Relevant**: entities that provide background information about the content of the document, but disregarding them would not affect negatively the comprehension of the document;
- **0 - Not Relevant/Not Mentioned**: any other entity in $\mathcal{E}$ that is not relevant or not mentioned in D.

The SE detection problem is to predict the saliency $\sigma(e|D)$ for each $e \in \mathcal{E}$. Note that the EL and SE problems are correlated and they almost coincide when a binary saliency function returning the relevance of an entity for D is adopted, i.e., $\sigma(e|D) = 1$ if $e \in \mathcal{E}_D$ and 0 otherwise.

The proposed *SEL* algorithm is able to discover $\mathcal{E}_D$, and in addition solves the SE problem, thus predicting $\sigma(e|D)$ for each $e \in \mathcal{E}_D$. The first step of *SEL* performs a *spotting* process, which detects potential entity mentions in the text. The hyperlink information of Wikipedia is exploited for this purpose. If the given document D contains a fragment of text s that is used as anchor text in Wikipedia to link to an entity e, then e is considered a *candidate entity* for the spot s. Since the same anchor text can be used in Wikipedia to reference any of several entities, a spot s might be associated with several candidate entities. The set of candidate entities can be very large, which makes it difficult to select the single correct entity for each spot, i.e., to disambiguate spots. However not all the possible entities are equally probable for a given spot, and candidate entities can be pruned to make the subsequent *disambiguation* step easier.

The first novelty in the proposed *SEL* algorithm is the usage of a machine-learned classifier with a set of easy-to-compute features to prune the candidate entities before disambiguation takes place. The goal of such classifier is to improve the precision of the state-of-the-art unsupervised techniques, without hindering recall: the classifier aims at filtering a small set of candidates without pruning any entity in $\mathcal{E}_D$. To train the classifier we investigated a novel and rich set of features, from which we selected only 8 *light* features.

The second step implements spot disambiguation. We devise two different solutions: the former aimed at solving the EL problem only, and the latter that, besides linking spots to correct entities, also scores them according to their saliency, thus combining the EL and SE discovery tasks. Also this step is based on machine-learning, this time using a regressor which is well suited for both the binary EL task (with a learned threshold value), or the multiclass SE problem.

The second novelty in the *SEL* algorithm is the blending of disambiguation and saliency prediction in a single step. We claim that this blending makes it possible to improve the accuracy of disambiguation for those spots/entities that are likely to be salient. The reason is that an EL task should not link everything, but just the relevant concepts, i.e., the

salient ones (thus excluding not relevant concepts, with a saliency score of 0). To learn an effective regressor for disambiguation, we analyzed a feature set wider than in the first step. By focusing on the relatively small number of candidate entities coming from the first step, it is possible to exploit complex and computationally *heavy* features, like those considering the entity relatedness graph.

3.1 Supervised Candidate Pruning

Potential entity mentions in a text are detected by exploiting the $\mathcal{KB}$: all the possible spots occurring in a given document D are matched against all the anchor texts and page titles in Wikipedia, and in case of an exact match (without any normalization on the text), a relationship is created between a spot s and the entities referred by s in Wikipedia.

Due to language ambiguity, the number of entities for each spot can be large. Formally, let $S_D = \{s_1, s_2, \ldots\}$ be the set of spots detected in D and $C_D = \{c_1, c_2, \ldots\}$, $C_D \subseteq \mathcal{E}$, the set of candidate entities, each of which is associated with some spot s_i. Indeed, the output of the spotting phase is a directed bipartite graph $G_D = (S_D, C_D, E_D)$, where E_D are the edges of the graph such that $(s_i, c_j) \in E_D$ if s_i is a text fragment used in Wikipedia for referring to entity $c_j \in \mathcal{E}$.

The goal of *Candidate Pruning* is to devise an effective entity pruning function ϕ: given a set of candidate entities C_D of the bipartite graph G_D identified by the spotting phase, ϕ finally produces a new set $C'_D = \phi(C_D)$, such that $|C'_D|$ is minimized and $|C'_D \cap \mathcal{E}_D|$ is maximized.

State-of-the-art algorithms perform a Heuristic Pruning (HP) of candidate entities C_D, by exploiting two measures, namely *commonness* and *link probability*, that can be precomputed as follows:

- The commonness of a candidate $c_j \in C_D$ for spot $s_i \in S_D$ is defined as the prior probability that an occurrence of an anchor s_i links to c_j. The commonness is a property of the edges of our bipartite graph. Given a spot $s_i \in S_D$, it is possible rank the outgoing edges and remove edges with low commonness.
- The link probability for a spot $s_i \in S_D$ is defined as the number of occurrences of s_i being a link to an entity in $\mathcal{KB}$, divided by its total number of occurrences in $\mathcal{KB}$. Therefore a spot with low link probability is rarely used as a mention to a relevant entity, and can be pruned from graph G_D.

Let τ_c and τ_{lp} be the *minimum commonness* and the *minimum link probability* (heuristic thresholds), it is possible to discard those graph edges with commonness lower than τ_c, and those spots with *link probability* lower than τ_{lp}. Note that when a spot s_i is pruned, also its outgoing edges are removed. After pruning the graph G_D on the basis of τ_c and τ_{lp}, some candidate entities in C_D may result disconnected from any spot, and they can thus be removed as well.

Setting a minimum threshold on commonness and link probability has been proven to be a simple and effective strategy, although heuristic, to limit the number of spots and associated candidate entities, without harming the recall of the EL process. Table 1 reports the performance of such heuristic pruning (HP) method over a well-known dataset (AIDA-CoNLL 2003 [7]) for different values of τ_c and τ_{lp}. The metrics adopted are precision (i.e., ratio of positive entities retained to the whole set of entities retained) and recall (i.e., ratio of positive entities retained to the whole set of positive entities). It is worth noting that commonly adopted thresholds ensure a good recall at the cost of a very low precision. The same table also reports the performance of the proposed solution, which is described below. For $\tau_c = 2\%$ the HP obtains up to 2% of improvement in recall with respect to the proposed method. On the other hand, with this setting the HP obtains a maximum precision of only 0.074, while the supervised solution achieves a precision of 0.367, i.e., 500% of improvement. Further experimental analysis is discussed in Section 4.2. Note that both Wikiminer [10] and Tagme [5] use $\tau_c = 2\%$, with the former using $\tau_{lp} = 6.5\%$ and the latter exploiting a more complex usage of the link probability value. In the following, we refer to the heuristic pruning strategy of Wikiminer as HP_W.

Table 1: Spotting performance for different values of τ_c and τ_{lp} (AIDA-CoNLL 2003 dataset).

Commonness	Link-Probability	Precision	Recall
0.005	0.02	0.022	0.907
0.005	0.03	0.025	0.900
0.005	0.04	0.029	0.893
0.005	0.05	0.036	0.893
0.01	0.02	0.032	0.891
0.01	0.03	0.038	0.884
0.01	0.04	0.043	0.877
0.01	0.05	0.052	0.877
0.02	0.02	0.048	0.864
0.02	0.03	0.056	0.856
0.02	0.04	0.063	0.850
0.02	0.05	0.074	0.850
0.04	0.02	0.072	0.839
0.04	0.03	0.082	0.831
0.04	0.04	0.092	0.826
0.04	0.05	0.103	0.826
Proposed *Candidate Pruning*		0.367	0.848

The *Candidate Pruning* method improves on the previous heuristic strategies by using a supervised technique. A binary classifier is learned to distinguish between relevant and irrelevant entities. Note that saliency has not taken into account in this step: a candidate entity c_j is considered relevant *iff* it is mentioned by the given document D. The training set is built from the ground truth on the basis of the bipartite graph $G_D = (S_D, C_D, E_D)$ generated by the spotting phase. A positive label is associated with $c_j \in C_D$ if $c_j \in \mathcal{E}_D$, and a negative label otherwise. Each entity $c_j \in C_D$ is represented with a large set of features extracted from the document, from the bipartite graph G_D and from the knowledge base $\mathcal{KB}$. These features are deeply discussed in Section 3.3. Eventually, only the candidate entities that are predicted to be relevant by the classifier are saved for the subsequent *Saliency Linking* step.

There are a couple of aspects relative to the ground truth that is worth discussing. First, class imbalance characterizes the training dataset, since on average we have that $|\mathcal{E}_D \cap C_D| \ll |C_D|$. Unfortunately a classifier learned from a training set with a strongly skewed class distribution may lead to poor performance. This is because most algorithms minimize the misclassification rate on the training set, hence favoring most frequent class, which in the specific case is the negative one. In order to deal with this issue, a cost model is introduced. Therefore, the classifier incurs a higher penalization when misclassifying an instance in a rare class.

Table 2: Light Features for Supervised Candidate Pruning: features are relative to a candidate entity c_j

1. **positions**	first, last, average, and standard deviation of the normalized positions of the spots referring to c_j
2. **first field positions**	document D is subdivided in 4 fields: *the title*, *the first three sentences*, *the last three sentences*, and *the middle sentences*; the normalized position of the first spot referring to c_j is computed for each field
3. **average position in sentences**	the average position of spots referring to c_j across the sentences of the document (salient entities are usually mentioned early)
4. **field frequency**	number of spots referring to c_j computed for each field of the document
5. **capitalization**	True *iff* at least one mention of c_j is capitalized
6. **uppercase ratio**	maximum fraction of uppercase letters among the spots referring to c_j
7. **highlighting**	True *iff* at least one mention of c_j is highlighted in bold or italic
8. **average lengths**	average term- and character-based length of spots referring to c_j
9. **idf**	maximum Wikipedia inverse document frequency among the spots referring to c_j
10. **tf-idf**	maximum document spot frequency multiplied by *idf* among the spots referring to c_j
11. **is title**	True *iff* at least one mention of c_j is present in the document title
12. **link probabilities**	maximum and average *link probabilities* of the spots referring to c_j
13. **is name/person**	True *iff* at least one mention of c_j is a common/person name (based on Yago – `http://goo.gl/glfBYN`)
14. **entity frequency**	total number of spots referring to c_j
15. **distinct mentions**	number of distinct mentions referring to c_j
16. **not ambiguity**	True *iff* at least one mention of c_j for which c_j is the only candidate entity
17. **ambiguity**	minimum, maximum and average ambiguity of the spots referring to c_j; spot ambiguity is defined as 1 minus the reciprocal of the number of candidate entities for the spot
18. **commonness**	maximum and average *commonness* of the spots referring to c_j
19. **max commonness × max link probability**	maximum *commonness* multiplied by the maximum *link probability* among the spots referring to c_j
20. **entity degree**	in-degree, out-degree and (undirected) degree of c_j in the Wikipedia citation graph
21. **entity degree × max commonness**	maximum *commonness* among the spots of c_j multiplied by the degree of c_j
22. **document length**	number of characters in D

Another key property which deserves attention concerns the choice of the feature space used to represent instances. Indeed, we distinguish between *light* and *heavy* features, i.e., either cheap or expensive to compute. We show that a small subset of these light features is able to generate a good classifier for the *Candidate Pruning*. The resulting classifier improves state-of-the-art heuristic techniques in terms of precision without hindering the recall, thus retaining most of the positive entities for the *Saliency Linking* step.

3.2 Supervised Saliency Linking

The *spotting* step in EL algorithms is always followed by a *disambiguation* phase: among the several candidates for a given spot, only one entity can be selected. The proposed *SEL* algorithm distinguish the following two tasks:

i) disambiguating spots also using contextual features, thus addressing the EL problem;

ii) predicting a saliency score for the relevant entities, thus addressing the EL and SE problem at the same time.

Both tasks are solved by learning a predictor of entity saliency. In the former case, an entity is considered relevant or irrelevant, i.e., $\sigma(e|D) \in \{0, 1\}$, while, in the latter, we have several degrees of relevance, i.e., $\sigma(e|D) \in \{0, 1, 2, 3\}$. The training dataset is built from the ground truth by considering only the candidate entities filtered by the *Candidate Pruning* step, and each entity c_j is labeled according to $\sigma(c_j|D)$. Note that all candidate entities c_k not mentioned in the document are labeled with $\sigma(c_k|D) = 0$.

This training dataset has two interesting properties. First, thanks to the *Candidate Pruning* step, the number of irrelevant entities is significantly reduced, and therefore the predictor is able to train on a quite balanced dataset with less noise. Second, by having a smaller number of candidate entities to deal with, it is possible to exploit more complex and powerful features able to better capture entity correlations. Indeed, besides the set of light features used in the *Candidate Pruning* step, an additional set of *heavy* features is added. These are mainly computed on the graphs induced by the Wikipedia hyperlinks, thus modeling the relationships among the candidate entities. It is worth remarking that this expensive feature extraction becomes feasible because the first step is able to strongly prune the original candidate set C_D. This new set of features is discussed in Section 3.3.

We remark that the *Saliency Linking* step implements disambiguation and saliency prediction at the same time. Disambiguation occurs implicitly as an incorrect entity c_k for a spot is predicted to have no saliency, i.e., $\sigma(c_k|D) = 0$. By tackling disambiguation and saliency prediction at the same time *SEL* achieves the goal of being accurate in linking the most relevant entities.

Note that during the *Saliency Linking* step the graph G_D is not considered, except via the features computed. When predicting the saliency of an entity, no information about the predicted saliency of other entities is exploited. Therefore, it is possible to have spots without any predicted relevant entity, and spots with more than one relevant entity. If needed, this can be easily fixed with a post-processing step not implemented in this work for the following reasons. First, it is much easier and clearer to consider the output of the *Saliency Linking* step as a flat set of entities, thus making it possible to easily adopt standard information retrieval measures, such as precision and recall. Second, it might be interesting in some application scenarios to have more than one annotation per spot, especially when more than one *facet* is relevant.

Table 3: Heavy features for Supervised Saliency Linking: most features are global and depend on the structure of the graph WG_D, others are specific for an entity

1. **graph size**	number of entities in WG_D
2. **graph diameter**	the diameter of WG_D
3. **node degree**	degree of given entity e in the undirected version of graph WG_D
4. **node average/median in-degree**	average and median node in-degree of WG_D
5. **node average/median out-degree**	average and median node out-degree of WG_D
6. **node average/median in-out-degree**	average and median node degree in the undirected version of graph WG_D
7. **farness**	the sum of the shortest paths lengths between entity e and all the other nodes in WG_D
8. **closeness**	the inverse of farness
9. **eigenvector centrality**	a measure of influence of a node in a network (Erkan and Radev [4])
10. **random walk**	the probability for a random walker to be at node e while visiting WG_D
11. **personalized random walk**	same as random walk, with a preference vector given by the entity frequencies in D
12. **graph cliques**	number of cliques in WG_D
13. **cross-cliques centrality**	a measure of connectivity of a node e in WG_D
14. **TAGME-like voting schema**	for each $e \in V_D$, we propose two normalizations of the TAGME-like voting schema: $\sum_{e' \in V_D \setminus \{e\}} \frac{Max\_comm(e') \cdot rel(e,e')}{Max\_ambig(e')}$ $\sum_{e' \in V_D \setminus \{e\}} \frac{Max\_comm(e') \cdot rel(e,e')}{\lvert V_D \rvert}$ where $rel(e, e')$ is the Milne and Witten relatedness function, whereas $Max\_ambig(e')$ and $Max\_comm(e')$ are defined in Table 2 (sections 16-17). Feature not dependent from WG_D.

3.3 Features

Given the candidate entities devised by the spotting phase in document D, the *SEL* algorithm represents with a vector of numerical features each candidate entity $c_j \in C_D$ in the bipartite graph $G_D = (S_D, C_D, E_D)$. Specifically, we distinguish between *light* features (i.e., cheap to be computed) which are generated for all $c_j \in C_D$, and *heavy* features (i.e., computationally expensive) which are computed only for the filtered candidate entities $C'_D = \phi(C_D) \subseteq C_D$, where $|C'_D| \ll |C_D|$.

Light features. Light features, illustrated in Table 2, are mainly derived from *attributes* associated with the mentions in S_D, which are then aggregated to build features for the mentioned entities. Some of them are computed on the basis of the occurrences of spots $s_i \in S_D$ within document D. For example, the positions of spots (1–3), their count (4), some typesetting features (5–7), their length (8). Features 9–10,12,18, rely instead on Wikipedia, but they are precomputed and stored in the dictionary used for spotting. We included features related to spots ambiguity, see 16–17. Finally, we included two novel features, 19 and 21, trying to blend together commonness, link probability and ambiguity signals.

Note that some of the features (2–4) explicitly refer to a semi-structure present in the dataset, with separate fields for different sections of each document. We exploited this semi-structure by distinguishing among spots occurring in the title of the document, in the first/last three sentences, and in the middle sentences. These features are aimed at exploiting information provided by the document structure.

Heavy features. These features are extracted for each candidate entity $c_j \in C'_D = \phi(C_D)$ to model the relationships among c_j and all the other entities in C'_D. To compute these features, specific subgraphs of Wikipedia graph are considered. Let $WG_D = (V_D, A_D)$ be one of such subgraphs, where both the set of vertices V_D and the set of arcs A_D can be defined in different ways:

Vertices V_D: the entities, i.e., Wikipedia nodes, identified by C'_D are extended with their neighborhoods in the Wikipedia graph. Two sets of vertices are exploited, denoted by V_D^0 and V_D^1: *i)* V_D^0 is simply equal to C'_D, as identified by our filtering step; *ii)* V_D^1 contains the vertices in V_D^0 extended with the entities associated with the Wikipedia pages that *link to* or are *linked by* entities in V_D^0.

Arcs A_D: three types of directed arcs are investigated: *i)* all the hyperlinks in Wikipedia between entities in V_D, considered as directed unweighted arcs. Therefore, we have two different sets of arcs, $A_D^0 \subset A_D^1$, one for each set of vertex sets $V_D^0 \subset V_D^1$; *ii)* the arcs derived from the Wikipedia hyperlinks, weighted by the Milne and Witten relatedness function [10], by pruning arcs whose relatedness is zero; *iii)* a weighted and undirected clique graph (i.e., each node is connected to each other), where edges are weighted by the Milne and Witten relatedness function. Also in this case, there are two sets of arcs $A_D^0 \subset A_D^1$. Finally, arcs with a weight below the median are discarded in order to preserve only the most important ones.

Heavy features, listed in Table 3, are computed on the 6 graphs resulting by the combination of the two vertex sets on the three edge sets described above. In total, each candidate entity is represented by a vector of 39 *light* features and 99 *heavy* features (16 features WG_D dependent times the 6 graphs, 2 from the TAGME-like scores and 1 the confidence score of the candidate pruning classifier at step 1).

It is worth remarking that the sets of vertices of WG_D (V_D^0 or V_D^1) are small enough to make the computation of these graph features feasible. This is due to the pruning capability of our first pruning step, which greatly reduces the size of the set of candidate entities.

4. EXPERIMENTS

4.1 Datasets

For the evaluation of EL performance we used the Test B part of the AIDA-CoNLL 2003 dataset [7]. This dataset contains a subset of news from Reuters Corpus V1 which were

Table 4: Agreement between groups of Expert (Exp) or Crowdflower (CF) annotators.

Annotators	Docs	Kendall's τ	Fleiss' κ	Kendall's τ *binary*	Fleiss' κ *binary*
CF *vs* CF	329	$0.54_{\pm.03}$	$0.33_{\pm.03}$	$0.68_{\pm.08}$	$0.49_{\pm.10}$
Exp *vs* Exp	62	$0.67_{\pm.11}$	$0.44_{\pm.14}$	$0.72_{\pm.03}$	$0.66_{\pm.04}$
CF *vs* Exp	62	$0.40_{\pm.06}$	$0.19_{\pm.03}$	$0.48_{\pm.09}$	$0.40_{\pm.08}$

manually linked to Wikipedia entities starting from candidates generated by the spotter of Aida [7]. The CoNLL dataset is composed of 231 documents with an average of 10.94 entities per document, hence resulting in $\approx 2,500$ mention to entities. Note that entities are not annotated with a saliency score. There exist other similar datasets such as the Knowledge Base Population track held by NIST Text Analysis Conference. However, the task is quite different as it requires annotating a given single mention in contrast to linking the full document, and it is released only with paid membership (free for the track participants).

In order to evaluate SE prediction performance, a human-assessed dataset of news was created and made publicly available, by relying on the Wikinews project[3]. Wikinews promotes the idea of participatory journalism, and provides a user-contributed repository of news. We chose this source for two main reasons: first, it is *open domain*, thus allowing us to redistribute the annotated dataset without the copyright constraints that affects similar datasets; second, because the news in Wikinews are already manually linked to entities of Wikipedia, thus making the dataset independent from the specific EL system used to detect entities. Due to some subjectivity in the assignment of a saliency score, each document (and thus also its entities) was annotated by multiple annotators, averaging the saliency scores.

An English dump of Wikinews containing news published from November 2004 to June 2014 was used, and the news that users linked to less than 10 or to more than 25 entities were filtered out. In addition, special news pages (e.g., News Briefs, or Wikinews shorts) were removed, as well as news longer than 2500 characters. The resulting dataset contains 604 news articles, uniform in text length and number of linked entities, each one with *title* and *body* fields.

Crowdflower[4], a crowd-sourcing platform, was then exploited for annotating linked entities with saliency scores. In order to get reliable human annotations, a *golden dataset* was created by asking to 4 expert annotators to provide entity saliency scores in a specific subset of 62 documents. Then, the Crowdflower quality control mechanisms allowed to use the golden dataset produced by the expert annotators to detect and ban malicious annotators. With a reward of 0.35$ per document, 400 documents (including the golden subset) were annotated by at least 3 different Crowdflower annotators in one week. Finally, documents where the annotators exhibited a low agreement were removed, obtaining the final Wikinews dataset, consisting of 365 annotated documents having an average of 12.02 entities per document, hence resulting in $\approx 4,400$ mentions to entities.

To evaluate the quality of the annotations we measured the Crowdflower annotators agreement with Fleiss' κ and Kendall's τ coefficients. The latter was measured by considering the ranked lists obtained by sorting the entities by the saliency label provided by the users. As reported in Table 4, we have $\kappa = 0.33_{\pm.03}$ and $\tau = 0.54_{\pm.03}$ among CrowdFlower users. The Fleiss' κ value suggests a *fair* agreement. This is due to the highly subjectivity of the task: different users may give different rates based on their experience, culture, etc. Our agreement results are however consistent with those reported in similar works [1]. Nevertheless, the Kendall's τ coefficient suggests a *good* ranking agreement. We also investigated agreement by collapsing *Highly Relevant* and *Partially Relevant* thus achieving a *binary* labeling. The agreement on such binary formulation is consistently higher, with $\kappa = 0.68_{\pm.08}$ and $\tau = 0.49_{\pm.10}$. This suggests that users agree in identifying *Top Relevant* entities, and they have slightly less agreement in discriminating between different degrees of relevance. Good agreement values were achieved also when comparing Crowdflower users with expert users.

[3] http://en.wikinews.org

[4] http://www.crowdflower.com

Table 5: Datasets description and spotting results.

	CoNLL	Wikinews
Documents	231	365
avg. $\lvert\mathcal{E}_D\rvert$	10.94	12.02
Top Relevant	—	436 (10%)
Highly Relevant	—	1685 (38%)
Partially Relevant	—	2261 (52%)
avg. $\lvert C_D\rvert$	549.54	790.05
avg. Max Rec = $\frac{\lvert C_D \cap \mathcal{E}_D\rvert}{\lvert\mathcal{E}_D\rvert}$	0.907	0.925

Finally, the different saliency labels provided by annotators were aggregated in order to have one unique saliency label per entity. The aggregation was achieved by averaging the annotators labels and by rounding the average value when a sharp classification is needed. The Wikinews dataset is publicly available and can be downloaded at the address http://dexter.isti.cnr.it/. Comparing with other datasets, we believe the annotations it provide are of high quality since it is not biased by users' queries to a search engine as in [11], and it does not rely on the naïve assumption, as in [3], that entities occurring in news abstract are salient while others are not salient.

Table 5 reports some statistics about the two dataset used in our experiments. Note that only 10% of the entities annotated in the Wikinews dataset are considered as *Top Relevant*. This suggests the importance of being able to detect the most salient entities in a document. We also report some statistics about the results of the Wikipedia-based spotter. The average number of candidate entities generated per document ranges between 500 and 800, corresponding to an average number of per-entity candidates of about 50 and 66 for the CoNLL and Wikinews datasets, respectively. These figures give a rough idea of the complexity of the disambiguation step. Altough the two datasets contain collectively ≈ 600 documents, they also contain a large number of mentions to entities, $\approx$ 6,900, which are essential in the creation and evaluation of the model, since the two phases are done on a per-entity basis.

The evaluation of the two steps of the *SEL* algorithms were carried out using *5-fold cross-validation* and averaging the results.

Table 6: Recall-oriented spotting performance.

	CoNLL			Wikinews		
	Rec	Prec	$\lvert C'_D\rvert$	Rec	Prec	$\lvert C'_D\rvert$
GBDT-$\mathcal{F}_l$	0.63	0.76	8.9	0.66	0.76	11.6
GBDT$_\omega$-$\mathcal{F}_l$	0.85	0.39	27.1	0.87	0.36	32.7
GBDT$_\omega$-$\mathcal{S}_l$	0.85	0.37	28.2	0.87	0.35	34.1
HP$_W$	0.85	0.07	127.4	0.91	0.06	171.9

4.2 Candidate Pruning Step

For each document D, a set of candidate entities C_D was generated with a dictionary based spotter, which exploits the Wikipedia anchors' text and article titles. This preliminary step generates an average of 549.54 and 790.05 candidate entities C_D for the CoNLL and Wikinews datasets respectively, as illustrated in Table 5.

To prepare the training set for a classifier used to prune C_D, a *positive* class label was associated to entities in $C_D \cup \mathcal{E}_D$, and a *negative* one to entities in $C_D \setminus \mathcal{E}_D$. It is worth remarking the *highly skewed* class imbalance. Indeed only 2% of $|C_D|$ are positive on CoNLL and 1.5% on Wikinews (see the corresponding sizes of $\mathcal{E}_D$ in Table 5).

An interesting information reported in Table 5 is the maximal recall achievable for the EL task, averaged over the set of documents in the given collection. This is smaller than 100% because a few positive entities in $\mathcal{E}_D$ were not detected by the spotter, that is $\mathcal{E}_D \cap C_D \neq \mathcal{E}_D$. This depends on the human annotation: in these cases annotators were able to recognize an entity in $\mathcal{KB}$ even if its mention in D is different from all the ones used in the $\mathcal{KB}$ and stored in our dictionary.

Table 6 shows the performance of the various pruning methods producing $C'_D = \phi(C_D)$. Note the column $|C'_D|$, which reports the *average* number of entities obtained after the pruning step, and compares its size with the original size $|C_D|$, reported in Table 5. The table also shows the Recall/Precision of the various methods in detecting the positive instances, i.e., the entities of C_D that are in $\mathcal{E}_D$.

In particular, Table 6 compares the heuristic pruning strategy HP$_W$ with the proposed supervised method. Indeed, the *Candidate Pruning* step adopts a state-of-the-art classification algorithm, the *Gradient Boosting Decision Tree* (GBDT) provided by the `scikit-learn` python library for machine learning. GBDT is trained on the light set of features $\mathcal{F}_l$. We denote this classifier by GBDT-$\mathcal{F}_l$.

Unfortunately, due to the severe class imbalance in the training set, the recall of GBDT-$\mathcal{F}_l$ is significantly worse than the baseline HP$_W$. This means that the classifier prunes too many positive entities. As expected, the precision of GBDT-$\mathcal{F}_l$ is better than the one obtained by HP$_W$, but its global performance is not satisfying. It is worth remarking that different settings of HP, not reported here, did not exhibit better performance in terms of precision.

We mitigated the issue of class imbalance by a re-balancing weight strategy, which re-weights the samples in the empirical objective function being optimized by the classifier. The weight given to each sample is inversely proportional to the frequency of its class in the training set. We denote by GBDT$_\omega$-$\mathcal{F}_l$ this new trained classifier, whose performance is very good. Its recall is similar to the one obtained by HP$_W$, but its precision is remarkably higher. By comparing the number of pruned candidate entities (column $|C'_D|$) with the non-pruned ones ($|C_D|$), the superior pruning power of the proposed method over HP$_W$ becomes apparent. Our supervised method is in fact able to prune $\approx 95\%$ of the initial set of candidates C_D, without hindering the recall.

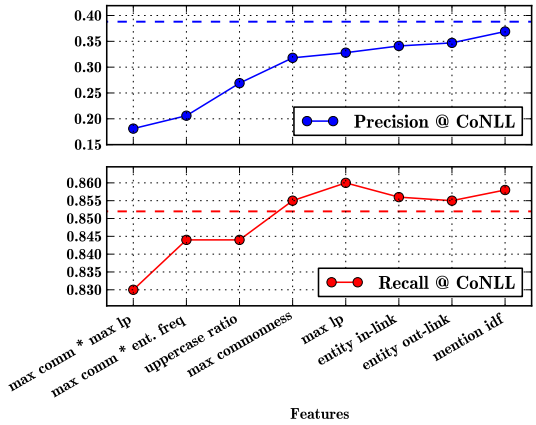

Figure 1: Incremental performance on step 1 using top k features.

The adopted GBDT implementation provides a standard measure of features' importance according to their contribution in optimizing the decision tree accuracy. We thus performed feature selection by considering the features sorted by importance, and trained a different classifier with the *top-k* features. Figure 1 shows the performance on the CoNLL dataset obtained by varying k up to the best 8 features. We denote this small set of top-8 features by $\mathcal{S}_l$. Note that the most important features are combinations of link probability, commonness, and entity frequency in Wikipedia. The performance of the classifier improves when we add further features. In fact, the performance of our GBDT$_\omega$-$\mathcal{S}_l$ classifier which employs the top-8 features, turned out to be very similar to the one of the classifier that employs the full set $\mathcal{F}_l$ (dashed line). This can also be observed by considering Table 6, where the performance of GBDT$_\omega$-$\mathcal{S}_l$ is reported for both CoNLL and Wikinews.

We conclude that the GBDT$_\omega$-$\mathcal{S}_l$ classifier provides the best performance on average for the two datasets, and that the *light* feature set $\mathcal{F}_l$ provides sufficient quality. Indeed, a smaller set of eight light features $\mathcal{S}_l$ suffices to train an effective classifier GBDT$_\omega$-$\mathcal{S}_l$, which is able to strongly prune the set of candidate entities, thus making feasible the subsequent step which needs to extract expensive graph-based features for each of these candidate entities.

4.3 Saliency Linking Step

In the second step, disambiguation and saliency prediction were performed by training a new model on the filtered set of candidates C'_D. In this case, the full feature set $\mathcal{F}$ was considered, including also an additional feature given by the confidence score of the candidate pruning classifier at step 1. The graph-based features are expensive to compute, but given the reduced number of entities per document, the computation is affordable.

In order to use the same model for both EL and SE tasks, we adopted a state-of-the-art regression algorithm, the *Gradient Boosting Regression Tree* (GBRT), again provided by

Table 7: Entity linking performance.

	CoNLL				Wikinews			
	Rec	Prec	F_1	P@3	Rec	Prec	F_1	P@3
GBRT-$\mathcal{F}$	0.76	0.71	0.72	0.82	0.76	0.72	0.72	0.88
GBRT$_\omega$-$\mathcal{F}$	0.73	0.74	0.72	0.81	0.77	0.70	0.72	0.85
GBRT-$\mathcal{S}_u$	0.71	0.71	0.69	0.80	0.74	0.71	0.71	0.85
Aida	0.76	0.72	0.73	0.82	0.66	0.73	0.68	0.80
Tagme	0.68	0.59	0.61	0.74	0.77	0.67	0.70	0.85
Wikiminer	0.55	0.43	0.46	0.65	0.78	0.53	0.62	0.87
Wikifier	0.52	0.33	0.36	0.43	0.41	0.34	0.36	0.35
Spotlight	0.48	0.30	0.32	0.46	0.56	0.31	0.38	0.54
1-Step GBRT-$\mathcal{F}_l$	0.69	0.69	0.67	0.81	0.70	0.73	0.69	0.86

the `scikit-learn` library, trained on the full set of features $\mathcal{F}$. The resulting model is denoted by GBRT-$\mathcal{F}$. A threshold was learned on the training set by optimizing the F_1 measure, and then used to filter out not relevant entities, i.e., having a score smaller than the learned threshold. The same linear search process was used for learning a filtering threshold on the confidence score for the competitors algorithms simply solving the EL problem.

To prove the benefits of the proposed two-steps algorithm, a regressor model trained on the original set of candidate entities C_D to predict the entity saliency (namely 1-Step GBRT-$\mathcal{F}_l$) was trained. This model exploited the light features $\mathcal{F}_l$ only, due to the high number of candidate entities, for which it was impossible to compute the heavy features.

The accuracy of the EL task was first analyzed by measuring precision, recall and F_1 score on the set of returned entities. The precision was also measured considering only the top-3 entities returned by the model, sorted by the annotation confidence for state-of-the-art algorithms or by the predicted score for our regression models. Note that, given the nature of the EL task, we are only interested in predicting relevant vs. irrelevant entities, resulting in the training of a binary model. Regarding the multi-class Wikinews dataset, all the positive scores were collapsed into a single relevant score. The distribution of positive and negative classes in $C'_D = \phi(C_D)$ became much more balanced after the pruning phase compared to the previous step (with a proportion of 35% / 65% respectively). Table 7 reports the EL performance for the various methods. In particular, state-of-the-art algorithms were compared with the proposed supervised method. The publicly available annotation service was used for each competitor algorithm except Wikifier, for which its available source code was used, with the best performing settings reported in the paper by the authors. The first two rows report the performance of the unbalanced model vs. the balanced one: since the dataset is only slightly unbalanced, they perform very similarly.

Also for this study, a subset of the top-10 most important features, denote as $\mathcal{S}_u$, was selected. The model trained using only this subset of features is GBRT-$\mathcal{S}_u$. It performs only slightly worse (-3% on F1 on CoNLL and -1% on Wikinews) than the model that uses all the features. Figure 2 reports the incremental F1 scores obtained by using this subset of features over CoNLL. It is worth noting that the top-2 features of this subset suffice to obtain performance higher then most state-of-the-art solutions. The most important features belong to different *families* of categories. We have

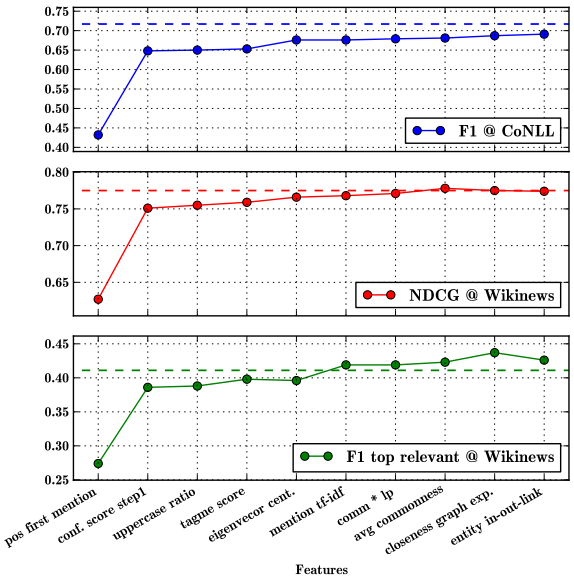

Figure 2: Incremental performance on step 2 using top k features.

some mention-based features (e.g., uppercase ratio or position first mention), some graph related features (e.g., eigenvector and Tagme-like) as well as features coming from the Wikipedia graph (e.g., entity degree) and the confidence score of the *Candidate Pruning* binary classifier.

The performance of the proposed solution were compared against state-of-the-art methods Aida, Spotight, Tagme, Wikiminer and Wikifier 2.0. The proposed full learned model obtained similar or even better performance when compared to the best performing algorithm on CoNLL (Aida) and Wikinews (Tagme), with an F1 of 0.72 on both the datasets. Indeed on Wikinews *SEL* exhibits +3% improvement on F1 compared to Tagme and +6% compared to Aida, while on CoNLL it performs only slightly worse than Aida (−1%) but it outperforms Tagme (+18%). It is worth noting that CoNLL dataset was created by using the Aida spotter, thus giving Aida an implicit advantage. Another interesting result is that it exhibits well balanced precision and recall values on both the datasets, while state-of-the-art competitors do not show a similar positive behaviour. Indeed, the proposed method shows the best performance on average across the two datasets for every measure adopted when using the full set of features, and it notably provides the best P@3 on average when using the feature set $\mathcal{S}_u$ only. Finally, some considerations about the 1-Step algorithm: despite its good performance, the method always performs worse than GBRT-$\mathcal{F}$ and GBRT-$\mathcal{S}_u$. It is worth noting that this single step algorithm provides EL annotations comparable or even better than most state-of-the-art algorithms. This confirms that entity saliency plays an important role as it also boosts entity linking methods. It is apparent that annotation confidence cannot approximate saliency.

Table 8 shows the saliency performance of the trained models. In this case the regressor makes use of all the saliency labels. For this experiment we used only the Wikinews dataset, since CoNLL is not annotated with the saliency. The performance on predicting the saliency was evaluated

Table 8: Saliency prediction performance on Wikinews.

	NDCG	Rec^{top}	Prec^{top}	F1^{top}
GBRT-$\mathcal{F}$	0.78	0.42	0.38	0.36
GBRT$_\omega$-$\mathcal{F}$	0.78	0.47	0.42	0.41
GBRT$_\omega$-$\mathcal{S}_u$	0.77	0.51	0.42	0.43
Aida	0.58	0.59	0.10	0.16
Tagme	0.65	0.45	0.13	0.18
Wikiminer	0.64	0.31	0.12	0.16
Wikifier	0.32	0.55	0.05	0.09
Spotlight	0.47	0.33	0.07	0.10
1-Step GBRT-$\mathcal{F}_l$	0.73	0.47	0.30	0.34

by using: i) the NDCG considering the entities sorted by saliency, in order to know how good is the function in ranking the entities by saliency, ii) Precision, Recall and F_1, considering only the most important entities, in order to know how good is our learned model in identifying the set of the *Top Relevant* entities (denoted as P^{top}, R^{top} and F_1^{top}). NDCG was measured on the set of entities selected by optimizing F_1 (as above), sorted by saliency/confidence score, whereas F_1^{top} is measured after optimizing a filtering threshold on the training data. It is worth recalling that state-of-the-art algorithms do not provide saliency scores, so we used the confidence scores as an indicator of how related are the entities to the document.

We observe that in this setting, the weighted model performs better than the unweighted one, since the distribution of the positive labels is not uniform. Moreover, the model that makes use of only the subset $\mathcal{S}_u$ of features has similar performance with respect to the model with all the features. As reported, *SEL* significantly outperforms the best performing state-of-the-art algorithm (Tagme) both in terms of NDCG and F_1^{top} with a relative improvement of +18% and +139% respectively.

We conclude that the recall-oriented pruning of the spotting results, along with the additional features extracted in the second step, provide a significant improvement over the 1-Step approach, with a substantial performance gap between the two models.

5. CONCLUSIONS

In this work we proposed a novel supervised Salient Entity Linking (*SEL*) algorithm that comprehensively addresses Entity Linking and Salient Entities detection problems. Besides improving Entity Linking performance with respect to state-of-the-art competitors, *SEL* predicts also the saliency of the linked entities. The algorithm exploits a two-step machine-learned process: first a *Candidate Pruning* step aimed at filtering out irrelevant candidate entities is performed, thus obtaining good precision figures without hindering recall; then, a *Saliency Linking* step effectively chooses the entities that are likely to be actually mentioned in the document and predicts their saliency.

The experiments conducted on two different datasets confirmed that the proposed solution outperforms state-of-the-art competitor algorithms in the Entity Linking task. In particular improvements in terms of F_1 of 6% w.r.t. Aida and 18% w.r.t. Tagme were measured. Moreover, *SEL* significantly outperforms the same competitors in the Salient Entities detection task of up to 18% and 139% in terms of NDCG and F_1^{top}, respectively. The latter analysis has been made possible thanks to the creation of a novel dataset of news manually annotated with entities and their saliency, hereinafter publicly available to the research community.

We believe that our comprehensive Entity Linking and Salient Entities detection approach constitutes a remarkable contribution to the field, since entity saliency detection is an important aspect of the whole document annotation pipeline, and salient entities should be weighted more than non salient ones in the evaluation of the annotation.

Acknowledgments. This work was partially supported by the EC H2020 Program INFRAIA-1-2014-2015 SoBigData: Social Mining & Big Data Ecosystem (654024).

6. REFERENCES

[1] R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, and T. Tran Duc. Repeatable and reliable search system evaluation using crowdsourcing. In *ACM SIGIR*, 2011.

[2] X. Cheng and D. Roth. Relational inference for wikification. In *Urbana*, 2013.

[3] J. Dunietz and D. Gillick. A new entity salience task with millions of training examples. In *EACL*, 2014.

[4] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. In *Journal Artificial Intelligence Res. (JAIR)*, 2004.

[5] P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *ACM CIKM*, 2010.

[6] M. Gamon, T. Yano, X. Song, J. Apacible, and P. Pantel. Identifying salient entities in web pages. In *ACM CIKM*, 2013.

[7] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *ACL EMNLP*, 2011.

[8] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: shedding light on the web of documents. In *ACM SEMANTiCS*, 2011.

[9] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *ACM CIKM*, 2007.

[10] D. Milne and I. H. Witten. Learning to link with wikipedia. In *ACM CIKM*, 2008.

[11] D. Paranjpe. Learning document aboutness from implicit user feedback and document structure. In *ACM CIKM*, 2009.

[12] F. Piccinno and P. Ferragina. From tagme to wat: a new entity annotator. In *Int. workshop on Entity recognition & disambiguation*, ACM SIGIR, 2014.

[13] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *ACL HLT*, 2011.

[14] H. Rode, P. Serdyukov, D. Hiemstra, and H. Zaragoza. Entity ranking on graphs: Studies on expert finding. 2007.

[15] W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. In *IEEE KDE*, 2015.