Computational Intelligence, Volume 000, Number 000, 2017

SEL: a Unified Algorithm for Salient Entity Linking

SALVATORE TRANI, CLAUDIO LUCCHESE, RAFFAELE PEREGO
National Research Council of Italy (CNR),
Institute of Information Science and Technologies (ISTI), Pisa, Italy

DAVID E. LOSADA
Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS),
Universidade de Santiago de Compostela, Spain

DIEGO CECCARELLI
Bloomberg LP, London, UK

SALVATORE ORLANDO
DAIS, Università Ca' Foscari Venezia, Italy

The *Entity Linking* task consists in automatically identifying and linking the entities mentioned in a text to their URIs in a given Knowledge Base. This task is very challenging due to natural language ambiguity. However, not all the entities mentioned in a document have the same 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 aimed at identifying a set of candidate entities that are likely to be mentioned in the document.The second step, 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. Furthermore, we employed *SEL* for Extractive Text Summarization. We found that entity saliency can be incorporated into text summarizers to extract salient sentences from text. The resulting summarizers outperform well-known summarization systems, proving the importance of using the *Salient Entities* information.

Key words: Entity Linking, Salient Entities, Machine Learning, Text Summarization.

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 (2007), 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 (Paranjpe, 2009) or *Salient Entities* (SE) discovery problem (Gamon et al., 2013), whose 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 following text: *"Maradona played his first World Cup*

tournament in 1982, when Argentina played Belgium in the opening game of the 1982 Cup in Barcelona". In the following, we show the annotations of this text performed by an EL algorithm that uses Wikipedia as a knowledge base:

> **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, 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* Mihalcea and Csomai (2007); Milne and Witten (2008). 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.

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

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.

This paper is an extension of Trani et al. (2016) where the *SEL* algorithm was first introduced. As original contributions of this paper, we enriched the experimental evaluation of *SEL* by discussing new results and insights. In particular, our analysis has been extensively conducted on two publicly available datasets by comparing the performance of our proposed method on both the Entity Linking and Saliency Detection tasks. Furthermore, we report on the exploitation of our entity saliency detection technology to feed novel text summarization techniques. SEL allows the entities that have high utility in understanding the topics of a document to be identified. The knowledge of such entities can be used to design novel extractive summarizers boosting the sentences mentioning the most salient entities in the document. We evaluated these new text summarizers on well-known single-document and multi-document summarization datasets, providing an empirical evidence of the positive effect of the salient-derived features.

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 rich and heterogeneous features used to represent entities within the machine learning algorithms adopted;
- novel single-document and multi-document summarizers that employ features based on entity-saliency to extract central sentences from documents.

In addition to the above technical contribution, we also contribute a novel dataset of news manually annotated with entities and their saliency, made publicly available to the research community to foster developments on this topic.

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 and Witten (2008) *relatedness*. 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 (2007) with *Wikify*. A substantial improvement has been the Wiki-Miner approach proposed by Milne and Witten (2008). It works by first identifying a set of non-ambiguous mentions and then using this set to disambiguate the ambiguous ones. Ferragina and Scaiella (2010) proposed an improved approach called Tagme, 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, Mendes et al. (2011) 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 (which is an extension of Ratinov et al. (2011)), Cheng and Roth (2013) 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, Hoffart et al. (2011) 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, Piccinno and Ferragina (2014) 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. Recently, a new research trend has emerged with preliminary investigation on the exploitation of features learned leveraging on Deep Neural Network (DNN) models. DNN are used to learn a semantic representation of words and entities in a unsupervised manner. These representations are exploited for estimating the relatedness among entities (Huang et al., 2015) or for improving the disambiguation phase (Yamada et al., 2016; Sun et al., 2015). Note that those solutions are limited to the EL problem while our work investigate how to jointly address both EL and SE. We thus leave to future work the usage of such DNN-based signals for improving the effectiveness of our algorithm. A thorough overview and analysis of the main approaches to EL and their evaluation is presented by Shen et al. (2015).

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* by Gamon et al. (2013) or *salient entity discovery* problem by Rode et al. (2007).

Gamon et al. (2013) studied the *aboutness* problem referred to the named entities occurring in Web pages. The approach used is partially inspired by Paranjpe (2009), 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 (2014) 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 by Rode et al. (2007), 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 $\mathbb{KB}$ be a knowledge base with a set of entities $\mathbb{E}$. The EL problem is to identify the entities $\mathbb{E}_D \subseteq \mathbb{E}$ 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 $\mathbb{E}$ that is not relevant or not mentioned in D.

The SE detection problem is to predict the saliency $\sigma(e|D)$ for each $e \in \mathbb{E}$. 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 $e \in \mathbb{E}_D$ and 0 otherwise.

The proposed *SEL* algorithm is able to discover $\mathbb{E}_D$, and in addition solves the SE problem, thus predicting $\sigma(e|D)$ for each $e \in \mathbb{E}_D$. 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 $\mathbb{E}_D$. 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 $\mathbb{K}B$: 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 $C_D = \{c_1,\ c_2, \ldots\}$, $C_D \subseteq \mathbb{E}$, 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 $c_j \in \mathbb{E}$.

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 $|C'_D \cap \mathbb{E}_D|$ 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

TABLE 1. Spotting performance for different values of τ_c and τ_{lp} on AIDA-CoNLL 2003 and Wikinews datasets. The heuristic pruning strategy of Wikiminer is highlighted with a light gray background.

		CoNLL		Wikinews	
Commonness	Link-Probability	Precision	Recall	Precision	Recall
0.005	0.02	0.022	0.907	0.016	0.925
0.005	0.03	0.025	0.900	0.020	0.922
0.005	0.04	0.029	0.893	0.024	0.919
0.005	0.05	0.036	0.893	0.027	0.915
0.005	0.065	0.044	0.892	0.033	0.909
0.01	0.02	0.032	0.891	0.026	0.921
0.01	0.03	0.038	0.884	0.031	0.917
0.01	0.04	0.043	0.877	0.036	0.915
0.01	0.05	0.052	0.877	0.041	0.911
0.01	0.065	0.063	0.876	0.050	0.905
0.02	0.02	0.048	0.864	0.040	0.915
0.02	0.03	0.056	0.856	0.048	0.911
0.02	0.04	0.063	0.850	0.056	0.909
0.02	0.05	0.074	0.850	0.062	0.906
0.02	0.065	0.089	0.849	0.074	0.900
0.04	0.02	0.072	0.839	0.060	0.908
0.04	0.03	0.082	0.831	0.072	0.904
0.04	0.04	0.092	0.826	0.083	0.901
0.04	0.05	0.103	0.826	0.092	0.898
0.04	0.065	0.121	0.826	0.109	0.893
Proposed *Candidate Pruning*		0.367	0.848	0.361	0.867

a link to an entity in $\mathbb{K}B$, divided by its total number of occurrences in $\mathbb{K}B$. 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 the well-known AIDA-CoNLL 2003 dataset released by Hoffart et al. (2011) and over a novel manually annotated dataset named Wikinews (see Section 5.1.1 for a description of the two datasets), 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 (Milne and Witten, 2008) and Tagme (Ferragina and Scaiella, 2010) 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. This strategy is highlighted in Table 1 with a light gray background.

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 $c_j \in \mathbb{E}_D$, 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 $|\mathbb{E}_D \cap C_D| \ll |C_D|$. 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. 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.

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:

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 $\times$ 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 $\times$ 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

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 (2008) relatedness function, 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. SUMMARIZATION

The saliency detection algorithm, which was described in the previous sections of this paper, is an effective solution for ranking the entities mentioned in a given text. This entity-based feature can be incorporated into text summarizers and exploited to extract salient sentences from text. This section is a report on our endeavors to inject entity-based features into standard text summarizers. Entities are core components of texts and they provide a great deal of information about the topics of the source texts.

Automatic Text Summarization is a powerful Text Mining technology that can rapidly digest and skim textual contents. Automatic summarizers are nowadays indispensable for dealing with increasing online data in a wide range of application domains (Mani, 2001). For instance, in web search, summaries –called *snippets*– are automatically built and attached to search engine hits. Automatic summarizers are also employed prominently for creating summaries of news stories, medical texts or biographical articles, just to name a few.

Summarization can be single-document or multi-document. Multi-document summarization is often more difficult because redundancy is a big issue in summarizing multiple texts. Summarization can also be query-biased or generic (also known as query-unbiased). Query-biased summaries are summaries that include query related content (e.g. web snippets). Generic summaries are not associated to a given query or topic, and provide a general sense of the information conveyed in the document(s).

Summarization is often done with abstractive or extractive methods. Abstractive summarizers are complex because they should have the ability of generating new sentences to convey the important information from textual documents. Synthesizing information and creating concise informative summaries following an abstractive approach is challenging, and requires extensive natural language processing. Furthermore, an abstractive summarizer should create coherent summaries that are easily readable and grammatically correct. There-

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 (2004))
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.

fore, the research community has focused more on extractive summarization (Gambhir and Gupta, 2017). In our work, we focus on single and multi-document extractive summarizers that produce generic summaries.

Extractive summarizers apply different methods to select salient parts of the source text. For example, cue words, position within the text, or centrality (estimated as the similarity to the centroid of the text) have been exploited for detecting salient extracts. Sentence-based summarizers identify the most important sentences in the source text and arrange them in some effective way. This involves three steps, namely: feature-based representation of sentences, sentence scoring, and summary creation by selecting sentences (Nenkova and McKeown, 2012). The first step often resorts to simplified representations of the sentences (e.g.,

bag of words and frequency-based weighting mechanisms), and content-based scores that estimate how central the sentence's words are. Other typical shallow features are location-based features. For instance, salient sentences tend to occur in certain specifiable positions within the text.

We claim that the most informative sentences might exhibit singular patterns of usage of entities and it might be the case that standard summarization features are unable to identify such patterns. We define here new entity-based sentence features for extractive summarization. These features are computed with *SEL* and then combined with standard sentence summarization features (position, centroid and length). This leads to a sentence scoring method that aggregates multiple types of evidence. Next, we proceeded to inject this sentence scoring method into a well-known summarization system that creates non-redundant summaries of the desired size. Finally, we performed single-document and multi-document summarization experiments and we analyzed the effects of the newly-derived features. These experiments are reported in subsection 5.2.

A wide range of features and summarization variants have been explored in the past. A full review of summarization can be found elsewhere (Gambhir and Gupta, 2017; Nenkova and McKeown, 2012; Mani, 2001). Furthermore, Ferreira et al. (2013) performed a quantitative and qualitative assessment of 15 sentence scoring algorithms. It is not our intention here to develop a state-of-the-art summarizer. We aim to take the first steps to understand the viability of saliency-based features to support extractive summarization. To meet this aim, we consider an open source, public domain, extractive summarizer as our main reference. As argued below, this well-known summarizer implements a number of standard summarization techniques and produces summaries in multiple ways. This extractive summarization system has been employed in a number of tasks such as summarization of mobile devices, web summarization, or novelty detection. As a matter of fact, it is a standard baseline in most summarization studies.

4.1. Summarization Approach

We estimate sentence importance by combining multiple types of evidence. For each candidate sentence, standard features, such as the position of the sentence in the source text, or the content-based similarity between the sentence and the document's centroid, are combined with entity-based features. First, we briefly describe some standard sentence features and the main components of the summarization system. Next, we present the new entity-based sentence features.

In many summarization cases, the sentences appearing at the beginning of a document provide much information about the topics of the document. Therefore, standard summarizers often weight the leading sentences more heavily. Centroid similarity is another standard feature commonly employed in summarization. This works as follows. Using standard statistics, a centroid is computed for each document to be summarized (e.g., a vector of tf-idf weights). This centroid tries to capture which words are central in the document. Following a similar approach, we obtain a weight-based representation for each candidate sentence. Finally, a similarity score (e.g., cosine similarity) between the weighted representation of the centroid and the weighted representation of the sentence is computed. This content-based matching approach favors sentences whose overall resemblance to the whole document is high.

MEAD (Radev et al., 2004) is a popular system that supports a variety of summarization strategies. It provides the implementation of effective baseline summarizers and, additionally, it has a flexible and modular architecture that permits to incorporate your own sentence features. MEAD supports single-document summarization (the input is a single document) and multi-document summarization (the input is a cluster of documents). The following

built-in features are automatically computed by MEAD and associated with each sentence of the document or cluster to be summarized[3]: Position, Centroid and Length. Position represents the position of the sentence in the document(s)[4]. Centroid is computed as the cosine overlap of the sentence with the centroid of the document (or cluster). Length is regarded as a cutoff feature: sentences whose length is below a given threshold are discarded. MEAD's aggregation module is based on linearly combining all feature weights and building a ranking of sentences by decreasing aggregated scores. This is an example of MEAD's sentence scoring approach for a summarizer that incorporates the three standard features:

$$score(sen) = \begin{cases} w_{cen} \cdot cen(sen) + w_{pos} \cdot pos(sen) & \text{if } len(sen) \geqslant thr_{len} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$cen(sen)$, $pos(sen)$ and $len(sen)$ are the values of Centroid, Position and Length for the sentence sen to be scored; w_{cen} and w_{pos} are the weights of the summarizer for Centroid and Position, and thr_{len} is the threshold for Length.

All sentences in the document (or cluster) are scored using this formula and a ranking by decreasing $score(sen)$ is built. Next, this initial ranking of sentences is re-ranked by a redundancy removal module. This module downgrades sentences that are too similar to sentences ranked above. In MEAD, the redundancy removal re-ranker offers a more diverse collection of sentences by implementing Maximal Marginal Relevance (MMR). A full description of MMR can be found in Carbonell and Goldstein (1998). Finally, the resulting ranking of sentences is employed to produce a summary of the desired size.

The standard sentence-based features described above have been enriched with several entity-based features, as to exploit the benefits of incorporating entity-derived information into text summarizers. We obtained these features by annotating each document independently from the others[5], and using the models trained on Wikinews for predicting the saliency of the linked entity. For single-document summarization we incorporated the following entity-based features:

- **SumSalMaxNorm**: sum of the predicted saliency of the entities annotated in each sentence. This sum was normalized into $[0, 1]$ by dividing by the maximum sum (computed across all sentences in the document).
- **SumSalLenNorm**: same as SumSalMaxNorm, but before normalizing by the maximum sentence score, a prior normalization is done by sentence length (so as to mitigate the advantage of long sentences above shorter ones).

For multi-document summarization we incorporated the following entity-based features:

- **SumAggSalMaxNorm**: the saliency score of each entity among the different documents is summed. This aggregation of scores leads to an overall estimation of entity saliency. This aggregated score is then used for summing the contribution of each entity to the sentence score, as described for the single-document feature SumSalMaxNorm. Finally, the sentence scores are normalized by their maximum score.
- **SumAggSalLenNorm**: same as SumAggSalMaxNorm, but adopting the prior normalization approach as described in SumSalLenNorm (i.e., by sentence length).
- **MaxAggSalLenNorm**: same as **SumAggSalMaxNorm**, but aggregating the entity saliency as the max of their predicted saliency.
- **TopSalientRelScoresMaxNorm**: using the top 3 salient entities of each document in the cluster, identify a subset of entities acting like a centroid. Then, sum the contribution

[3] All features range from 0 to 1.
[4] The first sentence gets a weight equal to 1 and the remaining sentences are assigned linearly decreasing weights.
[5] This also holds for multi-document summarization.

of each entity to the sentences where it appears in as the average relatedness between this entity and all the entities in the centroid set. The measure adopted for computing this similarity is the Milne and Witten (2008) *relatedness*. Finally, normalize the sentence scores by their maximum.

- **TopSalientRelScoresLenNorm**: As TopSalientRelScoresMaxNorm, but adopting the prior normalization approach as described in SumSalLenNorm (i.e., by sentence length).

We proposed also a slight variant of most of these features, identified by the postfix '_s2', where the contribution given by each entity is computed as the square of its predicted saliency. The main idea behind this variant is to boost sentences containing top salient entities.

5. EXPERIMENTS

This section reports the entity linking and saliency detection experiments (subsection 5.1) and the summarization experiments (subsection 5.2).

5.1. Entity linking and Saliency detection experiments

5.1.1. *Datasets.* For the evaluation of EL performance we used the Test B part of the AIDA-CoNLL 2003 dataset released by Hoffart et al. (2011). This dataset contains a subset of news from Reuters Corpus V1 which were manually linked to Wikipedia entities starting from candidates generated by the spotter of Aida (Hoffart et al., 2011). 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[6]. 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[7], 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

[6] http://en.wikinews.org
[7] http://www.crowdflower.com

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	$0.40_{\pm.06}$	$0.19_{\pm.03}$	$0.48_{\pm.09}$	$0.40_{\pm.08}$

subset of 62 documents. These annotations were collected using ELIANTO (Trani et al., 2014), an ad-hoc solution developed explicitly for accounting this problem and facilitating the creation of human assessed datasets with both entities and saliency. 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. This approach is similar to what was done by Lins et al. (2012).

To evaluate the quality of the annotations we measured the Crowdflower annotators agreement with Fleiss' κ (Fleiss, 1971) and Kendall's τ (Kendall, 1948) coefficients. The former is used to measure inter-rater reliability of agreement between a constant number of raters giving categorical ratings to a fixed number of items, while the latter is used to measure the rank correlation between two measured quantities and is based on the number of concordances and discordances in paired observations. The Kendall's coefficient 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 (Blanco et al., 2011). 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.

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 provides are of high quality since it is not biased by users' queries to a search engine as in Paranjpe (2009), and it does not rely on the naïve assumption, as in Dunietz and Gillick (2014), 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. Moreover, 61 documents out of 365 (17%) do not have any *Top Relevant* entity, indicating

TABLE 5. Datasets description and spotting results.

	CoNLL	Wikinews
Documents	231	365
avg. $\lvert\mathbb{E}_D\rvert$	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 $= \frac{\lvert C_D \cap \mathbb{E}_D\rvert}{\lvert\mathbb{E}_D\rvert}$	0.907	0.925

that: i) the linking done by the editors to the news is still not perfect, and ii) for several documents the problem originate from the lack of the entities in the $\mathbb{K}B$.

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 $\approx$ 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.

5.1.2. *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 $C_D \cup \mathbb{E}_D$, and a *negative* one to entities in $C_D \setminus \mathbb{E}_D$. It is worth remarking the *highly skewed* class imbalance. Indeed only 2% of $\lvert C_D\rvert$ are positive on CoNLL and 1.5% on Wikinews (see the corresponding sizes of $\mathbb{E}_D$ 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 $\mathbb{E}_D$ 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 $\mathbb{K}B$ even if its mention in D is different from all the ones used in the $\mathbb{K}B$ and stored in our dictionary.

Table 6 shows the performance of the various pruning methods producing $C'_D = \phi(C_D)$. Note the column $\lvert C'_D\rvert$, which reports the *mean* number of entities obtained after the pruning step, and compares its size with the original size $\lvert C_D\rvert$, 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 $\mathbb{E}_D$.

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 $\mathbb{F}_l$. We denote this classifier by GBDT-$\mathbb{F}_l$.

TABLE 6. Recall-oriented spotting performance.

	CoNLL			Wikinews		
	Rec	Prec	$\lvert C'_D \rvert$	Rec	Prec	$\lvert C'_D \rvert$
GBDT-$\mathbb{F}_l$	0.63	0.76	8.9	0.66	0.76	11.6
GBDT$_\omega$-$\mathbb{F}_l$	0.85	0.39	27.1	0.87	0.37	31.9
GBDT$_\omega$-$\mathbb{S}_l$	0.85	0.37	28.2	0.87	0.36	33.1
HP$_W$	0.85	0.09	124.1	0.90	0.08	169.5

Unfortunately, due to the severe class imbalance in the training set, the recall of GBDT-$\mathbb{F}_l$ is significantly worse than the baseline HP$_W$. This means that the classifier prunes too many positive entities. As expected, the precision of GBDT-$\mathbb{F}_l$ 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$_\omega$-$\mathbb{F}_l$ 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.

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 and Wikinews datasets obtained by varying k up to the best 8 features. We denote this small set of top-8 features by $\mathbb{S}_l$. 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$_\omega$-$\mathbb{S}_l$ classifier which employs the top-8 features, turned out to be very similar to the one of the classifier that employs the full set $\mathbb{F}_l$ (dashed line). This can also be observed by considering Table 6, where the performance of GBDT$_\omega$-$\mathbb{S}_l$ is reported for both CoNLL and Wikinews.

We conclude that the GBDT$_\omega$-$\mathbb{S}_l$ classifier provides the best performance on average for the two datasets, and that the *light* feature set $\mathbb{F}_l$ provides sufficient quality. Indeed, a smaller set of eight light features $\mathbb{S}_l$ suffices to train an effective classifier GBDT$_\omega$-$\mathbb{S}_l$, 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.

5.1.3. *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 $\mathbb{F}$ 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 the

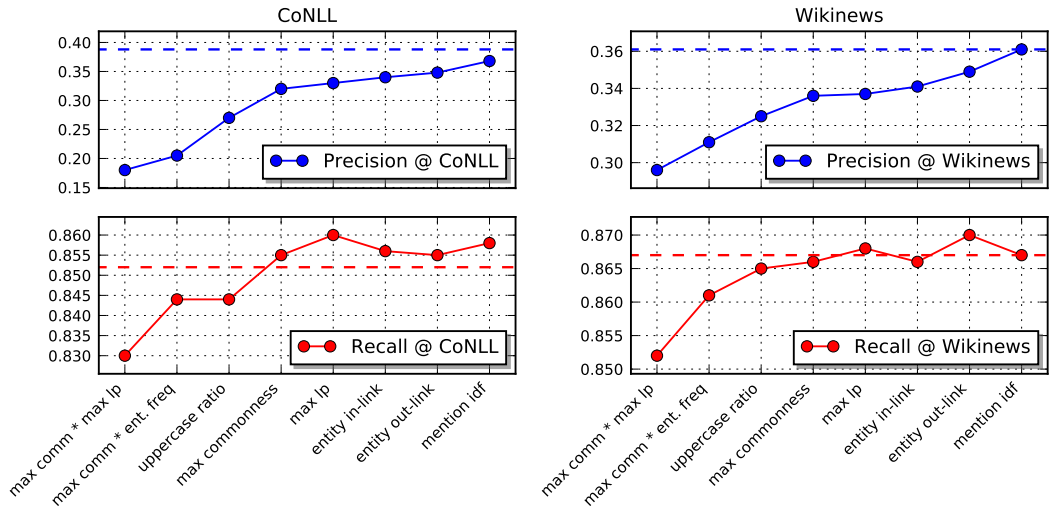

FIGURE 1. Incremental performance on step 1 using top k features.

`scikit-learn` library, trained on the full set of features $\mathbb{F}$. The resulting model is denoted by GBRT-$\mathbb{F}$. 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-$\mathbb{F}_l$) was trained. This model exploited the light features $\mathbb{F}_l$ 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 $\mathbb{S}_u$, was selected. The models trained using only this subset of features are GBRT-$\mathbb{S}_u$ and GBRT$_\omega$-$\mathbb{S}_u$, with the latter denoting the model that adopts the class imbalance solution. The two models perform very similarly each other, and only slightly worse (-4% on F1 on CoNLL and -1% on Wikinews) than the models that uses all the features. Figure 2 reports the incremental F1 scores obtained by using this subset of features over the two datasets. It is worth noting that the top-2 features of this subset suffice to obtain performance higher then most state-

TABLE 7. Entity linking performance.

	CoNLL				Wikinews			
	Rec	Prec	F_1	$P@3$	Rec	Prec	F_1	$P@3$
GBRT-$\mathbb{F}$	0.76	0.71	0.72	0.82	0.75	0.72	0.72	0.87
GBRT$_\omega$-$\mathbb{F}$	0.73	0.74	0.72	0.81	0.75	0.72	0.72	0.87
GBRT-$\mathbb{S}_u$	0.71	0.71	0.69	0.80	0.76	0.70	0.71	0.86
GBRT$_\omega$-$\mathbb{S}_u$	0.70	0.72	0.69	0.80	0.73	0.74	0.71	0.86
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-$\mathbb{F}_l$	0.69	0.69	0.67	0.81	0.70	0.73	0.69	0.86

of-the-art solutions. The most important features belong to different *families* of categories. We have 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,

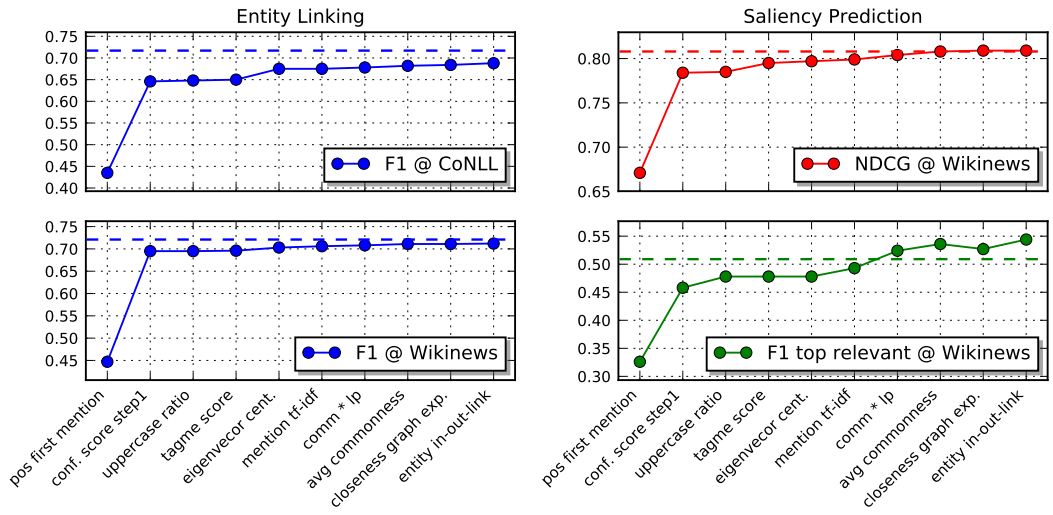

FIGURE 2. Incremental performance on step 2 using top k features.

TABLE 8. Saliency prediction performance on Wikinews.

	NDCG	Rec^{top}	Prec^{top}	F1^{top}
GBRT-$\mathbb{F}$	0.82	0.50	0.46	0.43
GBRT$_\omega$-$\mathbb{F}$	0.81	0.56	0.50	0.49
GBRT$_\omega$-$\mathbb{S}_u$	0.81	0.61	0.50	0.52
Aida	0.58	0.71	0.12	0.19
Tagme	0.65	0.54	0.16	0.22
Wikiminer	0.64	0.37	0.14	0.19
Wikifier	0.32	0.66	0.06	0.11
Spotlight	0.47	0.40	0.08	0.12
1-Step GBRT-$\mathbb{F}_l$	0.73	0.56	0.36	0.41

thus giving Aida an implicit advantage in terms of recall. 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 behavior. 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. Finally, some considerations about the 1-Step algorithm: despite its good performance, the method always performs worse than GBRT-$\mathbb{F}$ and GBRT-$\mathbb{S}_u$. 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 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 full set of entities, sorted by saliency/confidence score, whereas F_1^{top} is measured after optimizing a filtering threshold on the training data. To this purpose, the 61 documents without any *Top Relevant* entities has been discarded by the evaluation, so as to avoid misleading results. We highlight that no *state-of-the-art* algorithm provides saliency scores, therefore we used their annotation confidence as a proxy of entity saliency.

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 $\mathbb{S}_u$ of features has similar or even better 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 $+25\%$ and $+137\%$ respectively. Furthermore, Figure 2 reports the incremental F_1^{top} and NDCG scores obtained by using the subset $\mathbb{S}_u$ of features over the Wikinews dataset. It is worth noting that the model trained using only the top-7 features obtains performance similar to that of the full feature set $\mathbb{F}$, and by using all the top-10 features the model performs even better, with a $+6\%$ improvement in terms of F_1^{top}.

TABLE 9. Summarization collections used in our experiments

	Single-document summarization		
	DUC2001T	DUC2001	DUC2002
# documents	298	308	534
required summarization length	100 words	100 words	100 words
train/test	train	test	test

	Multi-document summarization		
	DUC2001MT	DUC2001M	DUC2002M
# clusters	30	29	116
avg # documents per cluster	9.97	10.17	9.59
required summarization length	100 words	100 words	100 words
train/test	train	test	test

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.

5.2. Summarization Experiments

We worked with several collections created under the Document Understanding Conference (DUC)[8]. We performed the following generic summarization tasks: i) single-document summarization (automatic summarization of a single news article), and ii) multi-document summarization (fully automatic summarization of multiple news articles on a single topic). Table 9 reports the main statistics of the collections and how we used them for training and testing. All documents are news articles obtained from the Text Retrieval Conference (TREC) and the average number of sentences per document is about 27.

The training step consisted only of learning the weights assigned to the new sentence features. We did not adapt the entity-based saliency estimation to the characteristics of these collections (we simply used the configuration learned from Wikinews).

Following existing practice, we evaluated the summarizers using ROUGE measures (Lin, 2004). This is a class of measures that automatically determine the quality of an automatic summary by comparing it to summaries created by humans (the DUC collections provide us with *manual* summaries for all documents and clusters). ROUGE measures the number of overlapping units (e.g., n-grams) between the automatic summary and the manual summary. ROUGE-2 and ROUGE-SU4 are two widely adopted ROUGE measures. ROUGE-2 is focused on counting bigram overlapping. ROUGE-SU4 counts overlapping of unigrams and *skip-bigrams* (bigram overlapping allowing for gaps with maximum length of 4).

We experimented with the following summarization methods:

- **standard MEAD**. This is the default MEAD configuration based on centroid, position and length. The default feature weights are 1, 1, and 9, respectively (meaning that sentences with less than 9 words are discarded and the remaining sentences are assigned an aggregated score equal to the sum of the centroid and position scores).

[8] http://duc.nist.gov.

- **lead-based MEAD**. This configuration of MEAD simply extracts the initial sentences of the document or cluster to build the summary.
- **random**. This is a naïve summarizer that randomly extracts sentences from the document or cluster.
- **MEAD + f_e** (where f_e is one of the entity-based features described above). This strategy consists of incorporating the feature f_e into *standard MEAD*. The weights and length threshold of the standard features are fixed to the default values (1, 1, and 9, respectively) and the weight of the new feature (e) is learnt by grid search on the training collection (the weights tested range from -1 to 1 in steps of 0.1). We optimized ROUGE-2. More sophisticated ways to optimize the weights can be implemented (e.g., Particle Swarm Optimisation, which was applied in Losada and Parapar (2016) for creating summarizers that work with dozens of features). However, we work here with a reduced set of features and focus on individually incorporating (and testing) each entity-based features. We leave sophisticated combinations and optimizations as future work.

5.2.1. *Results.* The experimental results obtained for the test collections are reported in Table 10 (single-document summarization) and Table 11 (multi-document summarization). The random summarizer performs poorly for both tasks. This is as expected, given its lack of sophistication.

Let us first focus on the results of single-document summarization. The inclusion of entity-based features on the top of standard MEAD led to improved summarizers. As a matter of fact, MEAD + f_e performs better than standard MEAD (for all f_e and for both performance measures). This suggests that the standard summarizer is unable to select sentences with prominent entities, and injecting entity-based features into this standard summarizer helps to create summaries with more salient entities (and more overlapping with gold summaries). For instance, SumSalLenNorm_s2, which is the best performing entity feature for single-document summarization, had assigned a weight of 1 during the training stage (the maximum in the range of our tuning grid: $[-1, 1]$). This means that the resulting summarizer (MEAD + SumSalLenNorm_s2) gives extra weight to sentences with salient entities (on the top of their Centroid or Position scores). The improvements of SumSalLenNorm_s2 over the other entity-based features give also credit to the way in which SumSalLenNorm_s2 mitigates the advantage of long sentences above shorter ones. Still, the overall results of single-document summarization do not give much support to entity-based features. The main reason is that a simple summarizer based on selecting the leading sentences leads to the highest ROUGE-2. Furthermore, the ROUGE-SU4 of MEAD + SumSalLenNorm_s2 is greater than the ROUGE-SU4 of lead-based MEAD but the improvement is tiny and statistically insignificant. The lead-based summarizer is a competitive solution for single-document summarization but it is the worst performing approach for multi-document summarization. When summarizing a single news article we can benefit from the style of writing of typical journalists, who express the main ideas first. However, summarizing a cluster of documents is a more difficult task where choosing the leading sentences from the clustered documents is ineffective.

Let us now discuss the results obtained for the multi-document summarization task. Standard MEAD is here the best performing baseline summarizer. It performs subsantially better than both the random summarizer and lead-based MEAD. Again, many entity-based features lead to improvements over standard MEAD; but SumAggSalMaxNorm and SumAggSalMaxNorm_s2 are the most promising features. SumAggSalMaxNorm features score the salient entities within the cluster of documents in an aggregated form. Each entity weight is based on aggregating how salient the entity is in every document of the cluster. This promotes entities that are central to the cluster. The results show that these features produce better multi-document summaries.

TABLE 10. Test results (Single-Document Summarization). The performance scores are reported together with 95% confidence intervals (in brackets). For each metric and collection the highest score is shown in boldface.

	ROUGE-2	ROUGE-SU4
DUC2001		
standard MEAD	.1793 (.1660, .1941)	.1813 (.1698, .1926)
random	.1277 (.1167, .1401)	.1420 (.1336, .1517)
lead-based MEAD	**.1931** (.1796, .2071)	.1825 (.1726, .1934)
MEAD + SumSalLenNorm	.1871 (.1735, .2016)	.1842 (.1729, .1955)
MEAD + SumSalLenNorm_s2	.1927 (.1789, .2068)	**.1902** (.1790, .2019)
MEAD + SumSalMaxNorm	.1852 (.1710, .2007)	.1839 (.1723, .1957)
MEAD + SumSalMaxNorm_s2	.1860 (.1719, .2016)	.1852 (.1737, .1971)
DUC2002		
standard MEAD	.1995 (.1912, .2080)	.1928 (.1855, .2000)
random	.1437 (.1357, .1520)	.1506 (.1441, .1573)
lead-based MEAD	**.2067** (.1986, .2154)	.1928 (.1862, .2000)
MEAD + SumSalLenNorm	.2039 (.1953, .2122)	.1976 (.1908, .2049)
MEAD + SumSalLenNorm_s2	.2046 (.1962, .2129)	**.1984** (.1915, .2056)
MEAD + SumSalMaxNorm	.2013 (.1929, .2096)	.1937 (.1866, .2004)
MEAD + SumSalMaxNorm_s2	.2035 (.1950, .2117)	.1965 (.1896, .2033)

Another interesting insight from our experiments is that all *s2* variants are better than their respective counterparts. This suggests that summarizers must focus on the top salient entities (rather than on marginally salient entities).

Attacking Text Summarization with entity-based features is a novel and interdisciplinary way of approaching the problem. We have provided preliminary empirical evidence on the effect of these features. Overall, our experiments suggest that entity-based features are meaningful and worth to be considered for Text Summarization. The improvements are modest but we think there is room for further enhancement. Observe that we did not adapt the saliency models to these DUC collections (we simply used the models learned on Wikinews) but, still, the results suggest that *SEL* can lead to improved summarizers (particularly for multi-document summarization). For single-document summarization, we only found modest improvements on ROUGE-SU4. In the future, we will further experiment with single-document summarization collections and we will try to confirm the effectiveness (or lack of) of entity-based features under different circumstances.

Observe also that this was a preliminary series of experiments and the aim of this evaluation was not to design a state-of-the-art summarizer. This would require combining evidence and features from multiple studies and summarization approaches. Instead, we focused on a well-known summarization system whose modular architecture permits to incorporate new features. These experiments allowed us to draw some initial conclusions about entity-based features in combination with some standard summarization features. But, of course, the role of entity-based features in enhancing state-of-the-art extractive summarizers requires further investigation.

TABLE 11. Test results (Multi-Document Summarization). The performance scores are reported together with 95% confidence intervals (in brackets). For each metric and collection the highest score is shown in boldface.

	ROUGE-2	ROUGE-SU4
DUC2001M		
standard MEAD	.0510 (.0374, .0646)	.0828 (.0682, .0986)
random	.0310 (.0213, .0424)	.0645 (.0544, .0747)
lead-based MEAD	.0303 (.0213, .0400)	.0639 (.0548, .0744)
MEAD + SumAggSalLenNorm	.0527 (.0378, .0697)	.0859 (.0714, .1022)
MEAD + SumAggSalLenNorm_s2	.0540 (.0408, .0681)	.0828 (.0683, .0989)
MEAD + SumAggSalMaxNorm	.0604 (.0445, .0775)	.0901 (.0762, .1055)
MEAD + SumAggSalMaxNorm_s2	**.0655** (.0483, .0841)	**.0925** (.0765, .1085)
MEAD + MaxAggSalLenNorm	.0466 (.0327, .0634)	.0790 (.0650, .0955)
MEAD + MaxAggSalLenNorm_s2	.0534 (.0405, .0680)	.0854 (.0715, .1009)
MEAD + TopSalientRelScoresLenNorm	.0510 (.0374, .0646)	.0828 (.0682, .0986)
MEAD + TopSalientRelScoresMaxNorm	.0587 (.0433, .0753)	.0873 (.0737, .1025)
DUC2002M		
standard MEAD	.0684 (.0610, .0769)	.0950 (.0870, .1032)
random	.0355 (.0301, .0413)	.0710 (.0659, .0764)
lead-based MEAD	.0433 (.0369, .0504)	.0659 (.0601, .0716)
MEAD + SumAggSalLenNorm	.0627 (.0554, .0704)	.0940 (.0870, .1012)
MEAD + SumAggSalLenNorm_s2	.0678 (.0596, .0762)	.0965 (.0893, .1037)
MEAD + SumAggSalMaxNorm	**.0708** (.0640, .0784)	.0970 (.0901, .1041)
MEAD + SumAggSalMaxNorm_s2	**.0708** (.0639, .0780)	**.0980** (.0914, .1050)
MEAD + MaxAggSalLenNorm	.0545 (.0483, .0607)	.0854 (.0792, .0920)
MEAD + MaxAggSalLenNorm_s2	.0607 (.0536, .0681)	.0892 (.0827, .0957)
MEAD + TopSalientRelScoresLenNorm	.0685 (.0610, .0769)	.0953 (.0873, .1035)
MEAD + TopSalientRelScoresMaxNorm	.0679 (.0605, .0753)	.0958 (.0890, .1034)

6. 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 25% and 137% 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 impacts on information extraction from text in a broader sense.

To experimentally assess this impact on a real use case, we investigated the usage of *SEL* to feed novel text summarization techniques. We thus exploited the entity saliency score predicted by *SEL* to design novel extractive summarizers boosting document sentences mentioning the most salient entities. The experiments conducted on several well-known summarization datasets provided the empirical evidence of the positive effect of including saliency-derived features in the summarization process. In particular we observed improvements in terms of ROUGE-SU4 of up to 5% on single-document datasets and up to 12% on multi-document datasets w.r.t. the Standard MEAD summarizer do not using saliency information. Overall, our results open a plenty of possibilities for solving many information extraction tasks making use of entity and saliency based information.

ACKNOWLEDGMENTS

This work was partially supported by the EC H2020 Program INFRAIA-1-2014-2015 SoBigData: Social Mining & Big Data Ecosystem (654024). David E. Losada thanks the financial support obtained from i) Ministerio de Economía y Competitividad of the Government of Spain and FEDER Funds under the research project TIN2015-64282-R, and ii) Consellería de Cultura, Educación e Ordenación Universitaria" (accreditation 2016-2019, ED431G/08) and the European Regional Development Fund (ERDF).

REFERENCES

BLANCO, ROI, HARRY HALPIN, DANIEL M HERZIG, PETER MIKA, JEFFREY POUND, HENRY S THOMPSON, and THANH TRAN DUC. 2011. Repeatable and reliable search system evaluation using crowdsourcing. *In* Proceedings of SIGIR, ACM, pp. 923–932.

CARBONELL, JAIME, and JADE GOLDSTEIN. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. *In* Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '98, ACM, New York, NY, USA. ISBN 1-58113-015-5. pp. 335–336. . http://doi.acm.org/10.1145/290941.291025.

CHENG, XIAO, and DAN ROTH. 2013. Relational inference for wikification. Urbana, **51**:61801.

DUNIETZ, JESSE, and DAN GILLICK. 2014. A new entity salience task with millions of training examples. *In* EACL 2014, pp. 205.

ERKAN, GÜNES, and DRAGOMIR R RADEV. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res.(JAIR), **22**(1):457–479.

FERRAGINA, P., and U. SCAIELLA. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). *In* Proceedings of CIKM, ACM, pp. 1625–1628.

FERREIRA, RAFAEL, LUCIANO DE SOUZA CABRAL, RAFAEL DUEIRE LINS, GABRIEL PEREIRA E SILVA, FRED FREITAS, GEORGE DC CAVALCANTI, RINALDO LIMA, STEVEN J SIMSKE, and LUCIANO FAVARO. 2013. Assessing sentence scoring techniques for extractive text summarization. Expert systems with applications, **40**(14):5755–5764.

FLEISS, JOSEPH L. 1971. Measuring nominal scale agreement among many raters. Psychological bulletin, **76**(5):378.

GAMBHIR, MAHAK, and VISHAL GUPTA. 2017. Recent automatic text summarization techniques: a survey. Artificial Intelligence Review, **47**(1):1–66. ISSN 1573-7462. . http://dx.doi.org/10.1007/s10462-016-9475-9.

GAMON, MICHAEL, TAE YANO, XINYING SONG, JOHNSON APACIBLE, and PATRICK PANTEL. 2013. Identifying salient entities in web pages. *In* Proceedings of CIKM, ACM, pp. 2375–2380.

HOFFART, JOHANNES, MOHAMED AMIR YOSEF, ILARIA BORDINO, HAGEN FÜRSTENAU, MANFRED

PINKAL, MARC SPANIOL, BILYANA TANEVA, STEFAN THATER, and GERHARD WEIKUM. 2011. Robust disambiguation of named entities in text. *In* Proceedings of EMNLP, Association for Computational Linguistics, pp. 782–792.

HUANG, HONGZHAO, LARRY HECK, and HENG JI. 2015. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *In* arXiv preprint arXiv:1504.07678.

KENDALL, MAURICE GEORGE. 1948. Rank correlation methods.

LIN, CHIN-YEW. 2004. ROUGE: A package for automatic evaluation of summaries. *In* Text Summarization Branches Out: Proceedings of the ACL-04 Workshop. *Edited by* S. S. Marie-Francine Moens. Association for Computational Linguistics, Barcelona, Spain, pp. 74–81.

LINS, RAFAEL DUEIRE, STEVEN J SIMSKE, L DE SOUZA CABRAL, G DE SILVA, RINALDO LIMA, RAFAEL F MELLO, and LUCIANO FAVARO. 2012. A multi-tool scheme for summarizing textual documents. *In* Proc. of 11st IADIS International Conference WWW/INTERNET 2012, pp. 1–8.

LOSADA, DAVID E., and JAVIER PARAPAR. 2016. Injecting multiple psychological features into standard text summarisers. *In* Proceedings of the 4th Spanish Conference on Information Retrieval, CERI '16, ACM, New York, NY, USA. ISBN 978-1-4503-4141-7. pp. 3:1–3:8. . http://doi.acm.org/10.1145/2934732.2934734.

MANI, INDERJEET. 2001. Automatic Summarization. John Benjamins Publishing Company.

MENDES, PABLO N, MAX JAKOB, ANDRÉS GARCÍA-SILVA, and CHRISTIAN BIZER. 2011. Dbpedia spotlight: shedding light on the web of documents. *In* Proceedings of SEMANTiCS, ACM, pp. 1–8.

MIHALCEA, RADA, and ANDRAS CSOMAI. 2007. Wikify!: linking documents to encyclopedic knowledge. *In* Proceedings of CIKM, ACM, pp. 233–242.

MILNE, DAVID, and IAN H WITTEN. 2008. Learning to link with wikipedia. *In* Proceedings of CIKM, ACM, pp. 509–518.

NENKOVA, ANI, and KATHLEEN MCKEOWN. 2012. A survey of text summarization techniques. *In* Mining Text Data. *Edited by* C. C. Aggarwal and C. Zhai. Springer, pp. 43–76. ISBN 978-1-4419-8462-3.

PARANJPE, DEEPA. 2009. Learning document aboutness from implicit user feedback and document structure. *In* Proceedings of CIKM.

PICCINNO, FRANCESCO, and PAOLO FERRAGINA. 2014. From tagme to wat: a new entity annotator. *In* Proceedings of the first international workshop on Entity recognition & disambiguation, ACM, pp. 55–62.

RADEV, DRAGOMIR, TIMOTHY ALLISON, SASHA BLAIR-GOLDENSOHN, JOHN BLITZER, ARDA ÇELEBI, STANKO DIMITROV, ELLIOTT DRABEK, ALI HAKIM, WAI LAM, DANYU LIU, JAHNA OTTERBACHER, HONG QI, HORACIO SAGGION, SIMONE TEUFEL, MICHAEL TOPPER, ADAM WINKEL, and ZHU ZHANG. 2004. MEAD – A platform for multidocument multilingual text summarization. *In* Conference on Language Resources and Evaluation (LREC), Lisbon, Portugal.

RATINOV, LEV, DAN ROTH, DOUG DOWNEY, and MIKE ANDERSON. 2011. Local and global algorithms for disambiguation to wikipedia. *In* Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, pp. 1375–1384.

RODE, HENNING, PAVEL SERDYUKOV, DJOERD HIEMSTRA, and HUGO ZARAGOZA. 2007. Entity ranking on graphs: Studies on expert finding.

SHEN, WEI, JIANYONG WANG, and JIAWEI HAN. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. Knowledge and Data Engineering, IEEE Transactions on, **27**(2):443–460.

SUN, YAMING, LEI LIN, DUYU TANG, NAN YANG, ZHENZHOU JI, and XIAOLONG WANG. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. *In* IJCAI, pp. 1333–1339.

TRANI, SALVATORE, DIEGO CECCARELLI, CLAUDIO LUCCHESE, SALVATORE ORLANDO, and RAFFAELE PEREGO. 2014. Manual annotation of semi-structured documents for entity-linking. *In* Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ACM, pp. 2075–2077.

TRANI, SALVATORE, DIEGO CECCARELLI, CLAUDIO LUCCHESE, SALVATORE ORLANDO, and RAFFAELE PEREGO. 2016. Sel: A unified algorithm for entity linking and saliency detection. *In* Proceedings of the 2016 ACM Symposium on Document Engineering, ACM, pp. 85–94.

YAMADA, IKUYA, HIROYUKI SHINDO, HIDEAKI TAKEDA, and YOSHIYASU TAKEFUJI. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. *In* arXiv preprint arXiv:1601.01343.