Computational Intelligence, Volume 000, Number 000, 2018

On Detecting Salient Wikipedia Entities in Texts

MARCO PONZA, PAOLO FERRAGINA, FRANCESCO PICCINNO
University of Pisa
{name.lastname}@di.unipi.it

We study the problem of entity salience by proposing the design and implementation of SWAT, a system that identifies the salient Wikipedia entities occurring in an input document. SWAT consists of several modules that are able to detect and classify on-the-fly Wikipedia entities as salient or not, based on a large number of syntactic, semantic and latent features properly extracted via a supervised process which has been trained over millions of examples drawn from the New York Times corpus. The validation process is performed through a large experimental assessment, eventually showing that SWAT improves known solutions over all publicly available datasets. We release SWAT via an API that we describe and comment in the paper in order to ease its use in other software.

Key words: Entity Linking; Entity Salience; Information Retrieval; Machine Learning; Natural Language Processing; Wikipedia

1. INTRODUCTION

Detecting *salient* information in documents, such as keywords (Bruza and Huibers, 1996; Paranjpe, 2009; Hasan and Ng, 2014), sentences (Mihalcea and Tarau, 2004) or Wikipedia entities (Dunietz and Gillick, 2014; Trani et al., 2017), has become a fundamental task on which different Information Retrieval (IR) and Natural Language Processing (NLP) tools hinge upon to improve their performance, such as contextual ads-matching systems (Radlinski et al., 2008), exploratory search (Anick, 2003), document similarity (Gabrilovich and Markovitch, 2007; Ni et al., 2016), web search ranking (Gamon et al., 2013; Schuhmacher et al., 2015) and news suggestion (Fetahu et al., 2015).

In this paper we propose a new system called SWAT (**S**alient **W**ikipedia **A**nnotation of **T**ext), which constitutes the state-of-the-art in detecting salient Wikipedia entities occurring in an input text. The software architecture of SWAT relies on a pipeline organized in three main modules: *Document Enrichment*, *Feature Generation* and *Entity Salience Classification*. Given an input document, the *Document Enrichment* module annotates it with proper syntactic, semantic and latent features that are automatically extracted through the deployment of four software components: (i) CORENLP (Manning et al., 2014), the most well-known NLP framework to analyze the grammatical structure of sentences, is used to extract the morphological information coming from the dependency trees built over the sentences of the input document; (ii) WAT (Piccinno and Ferragina, 2014), one of the best publicly available entity linkers (Usbeck et al., 2015), is used to annotate the text with proper Wikipedia entities and to build an entity graph for weighting the importance of these entities and their semantic relationships; (iii) TEXTRANK (Mihalcea and Tarau, 2004), the popular document summarizer, is used to return a keyphrase score for

The present paper is an extended version of the one published in the *Proceedings of the 22nd International Conference on Natural Language & Information Systems (NLDB, 2017)* (Ponza et al., 2017).

each sentence of the input document; and (iv) WORD2VEC, the continuous vector space representation of words and entities captured via deep neural networks, is used to enrich the entity graph of point (ii) with distributional semantic features. Subsequently, the *Feature Generation* module dispatches the enriched information generated from the first stage to a number of other software components in order to map each entity into its proper vector of features, which significantly expands the ones investigated in previous papers (Dunietz and Gillick, 2014; Trani et al., 2017). Finally, these feature vectors are given in input to the *Entity Salience Classification* module that leads to discriminate entities into salient and non-salient.

The validation of our system is performed through a large experimental assessment executed over two datasets, known as New York Times and Wikinews. SWAT will be compared against two systems that constitute the state-of-the-art in this setting, namely CMU-GOOGLE (Dunietz and Gillick, 2014) and SEL (Trani et al., 2017). This experimental study will show that SWAT raises the best known performance in terms of F1 up to about 13.5% (absolute) over CMU-GOOGLE system and up to 6.3% (absolute) over SEL system in either of the two experimented datasets. These F1-results will be complemented with a throughout discussion about the impact of each feature (old and new ones) onto the overall performance of our system and on how the position of salient entities does influence the efficacy of their detection. In this latter setting, we will show that the improvement of SWAT with respect to CMU-GOOGLE over the largest dataset New York Times may get up to 14% in micro-F1.

In summary, the main contributions of the paper are the following ones:

- We design and implement SWAT, an effective entity salience system that detects the salient entities of a document via novel algorithms that extract a rich set of features: syntactic (sentences' ranking, dependency trees, etc.), latent (i.e. word and entity embeddings), and semantic (computed via a new graph representation of entities and several centrality measures). Despite the use of word and entity embeddings is not new in IR, we are the first (to the best of our knowledge) to investigate its effectiveness on the entity salience task with a proper engineering of features based on these latent representations of entities.
- We are also the first ones to offer an extensive experimental comparison among all known entity salience systems (i.e. SWAT, SEL and CMU-GOOGLE, plus several other baselines) over the available datasets: i.e., New York Times (Dunietz and Gillick, 2014) and Wikinews (Trani et al., 2017).
- These experiments show that SWAT consistently improves the F1 performance of CMU-GOOGLE and SEL over those two datasets by achieving, respectively, an improvement of about +11% (absolute) and +5% (absolute).
- These figures are accompanied by a thoughtful analysis of SWAT's features, efficiency and errors, thus showing that all of its components are crucial to achieve its improved performance both in F1 and time efficiency.
- In order to encourage the development of other research built upon entity salience tools, we release SWAT as a public API[1], which actually implements the full entity linking-and-salience pipeline thus ease its plugging into other software.

The paper is organized as follows. Section 2 discusses the problem of the detection of

[1] `https://sobigdata.d4science.org/web/tagme/swat-api.`

salient information in texts by presenting known solutions and their limitations. Section 3 describes the design principles at the core of SWAT by detailing its three main constituting modules and posing particular attention on the sophisticated and novel feature extraction process, and on the design of the API and GUI interface which makes SWAT easily pluggable into any other IR tool (see Section 2). Section 4 then digs into the experimental comparison between SWAT and the current state-of-the-art systems CMU-GOOGLE (Dunietz and Gillick, 2014) and SEL (Trani et al., 2017) over the New York Times and Wikinews datasets. The experimental figures will show a coherent and significant improvement of SWAT with respect to those systems on both datasets. The subsequent Section 5 will extend the previous experimental analysis with a discussion on four engineering and algorithmic aspects pertaining with the design of SWAT: (i) the impact that the features have on the quality of its entity salient predictions, (ii) its efficiency in terms of constituting modules and used features, (ii) the impact of the training-set size onto its generalization ability and, finally, (iv) a thoughtful error analysis that will highlight the deficiencies of the known datasets. Taking inspiration from the previous detailed discussion, Section 6 will introduce few interesting research directions which would be worth to be investigated in the near future because could lead to further improvements on the solution to the entity salience task.

2. RELATED WORK

Classical approaches for detecting salient information in documents are known under the umbrella topic of *keyphrase extraction* (Hasan and Ng, 2014). These systems identify keyphrases through the *lexical* elements of the input text, such as words labeled with specific POS-tags (Mihalcea and Tarau, 2004; Liu et al., 2010; Gamon et al., 2013), n-grams (Turney, 2000) or words that belong to a fixed dictionary of terms (Paranjpe, 2009). The salient keyphrases are then selected from these lexical elements via supervised or unsupervised machine-learning models (Paranjpe, 2009; Gamon et al., 2013). Unfortunately, keyphrase extraction systems commonly incur in several limitations which have been properly highlighted in the previous literature (e.g. see (Hasan and Ng, 2014)): (i) their interpretation is left to the reader (aka, *interpretation errors*); (ii) words that appear frequently in the input text often induce the selection of not-salient keyphrases (aka, *over-generation errors*); (iii) infrequent keyphrases go undetected (aka, *infrequency errors*); and (iv) by working at a pure lexical level the keyphrase-based systems are unable to detect the semantic equivalence between two keyphrases (aka, *redundancy errors*).

Given these limitations, some researchers tried very recently to introduce some "semantics" into the salient representation of a document by taking advantage of the advances in the design of entity linkers (e.g. see (Usbeck et al., 2015) and the references therein). The key idea underlying those approaches consists of identifying in the input text meaningful sequences of terms and link them to *unambiguous* entities drawn from a Knowledge Base, such as Wikipedia, DBpedia (Bizer et al., 2009), Freebase (Bollacker et al., 2008), Wikidata (Vrandečić and Krötzsch, 2014), YAGO (Suchanek et al., 2007) or BabelNet (Navigli and Ponzetto, 2012). Since those entities occur as nodes in a graph, new and more sophisticate methods have been designed that empower classic approaches and thus allow to achieve better solutions for many well-known problems formulated over microblogging (Ferragina et al., 2015; Liu et al., 2013; Meij et al., 2012), text classification and clustering (Scaiella et al., 2012; Vitale et al., 2012), Knowledge Base construction (Niu et al., 2012; Bovi et al., 2015; Nguyen et al., 2017) and query understanding (Blanco et al., 2015; Cornolti et al., 2016).

Assigning a proper *salient* label to Wikipedia entities is still in its infancy and, indeed, only two approaches are known: namely, the CMU-GOOGLE (Dunietz and Gillick, 2014) system and the SEL (Trani et al., 2017) system. The first one uses a proprietary entity linker to extract entities from the input text and a binary classifier based on very few and simple features to distinguish between salient and non-salient entities. Authors (Dunietz and Gillick, 2014) have shown that their system significantly outperforms a simple baseline via some experiments executed over the large and well-known New York Times dataset. Unfortunately, the software deploys proprietary modules that make it publicly unavailable. In the end the authors of (Dunietz and Gillick, 2014) concluded that: *"There is likely significant room for improvement, [...]. Perhaps features more directly linked to Wikipedia, as in related work on keyword extraction, can provide more focused background information"*.

Following this intuition, Trani *et al.* (Trani et al., 2017) proposed the second known approach, called SEL, that hinges on a supervised two-step algorithm comprehensively addressing both entity linking (to Wikipedia's articles) and entity salience detection. The first step is based on a classifier aimed at identifying a set of candidate entities that are mentioned in the document, thus maximizing the precision without hindering the recall; the second step is based on a regression model that aims at scoring those candidate entities. Unfortunately SEL was compared only against *pure* entity linkers (such as TAGME (Ferragina and Scaiella, 2012)), which are not designed for the entity salience task, the system is yet publicly unavaiable and, furthermore, its experimental figures were confined to a new dataset (i.e. Wikinews), which was much smaller than NYT, and thus missed a comparison against the CMU-GOOGLE system.

As a result, the two entity salience systems above are not publicly available and their experimental figures are incomparable. In the present paper we continue the study of the entity salience problem by introducing a novel system, that we call SWAT, whose main goal is to efficiently and efficaciously address these open issues by improving the state of the art.

3. SWAT: A NOVEL ENTITY SALIENCE SYSTEM

In this section we describe our system SWAT, which aims at identifying the salient Wikipedia entities of an input document through a pipeline of three main modules: *Document Enrichment*, *Feature Generation* and *Entity Salience Classification*. Figure 1 provides a high-level overview of SWAT.

Document Enrichment. The first module aims at enriching the input document d with a set of semantic, morphological, syntactic and latent information. Specifically, this module is organized in four main components:

(1) CORENLP (Manning et al., 2014) is the component in charge of enriching the document with proper morphological NLP annotations. Specifically, it tokenizes the input document d, assigns a part-of-speech-tag to each token, generates the dependency relations between the tokens, identifies proper nouns and finally generates the coreference chains.
(2) TEXTRANK (Mihalcea and Tarau, 2004) is a component that works by taking as input the sentences tokenized by CoreNLP and by rank them via a random walk over a complete graph in which nodes are sentences and the weights of the edges

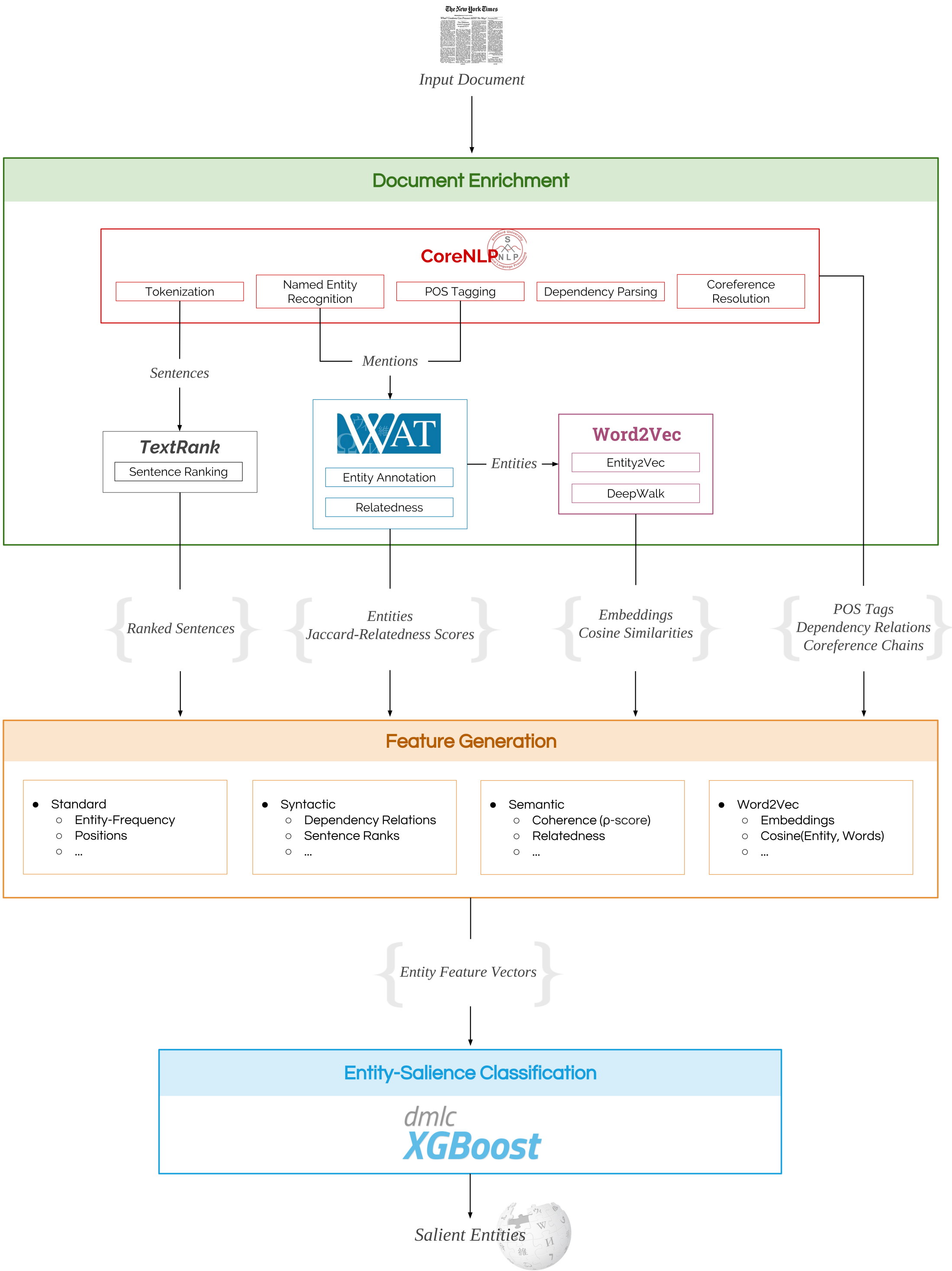

FIGURE 1: Three-module architecture of SWAT.

are computed as a function of the normalized number of common tokens between the connected sentences.

(3) WAT (Piccinno and Ferragina, 2014) is the component that aims to enrich d with a set of semantic annotations (m, e), where m is a sequence of words (i.e. *mentions*, provided by CoreNLP as proper or common nouns) and e is an entity (i.e. Wikipedia page). Specifically, WAT disambiguates every mention m by assigns to each mention an entity provided with two main scores:

   (a) *commonness*, which represents the probability that m is disambiguated by e;

   (b) *coherence* (denoted by ρ), which represents the semantic coherence between the annotation and its textual context in d.

   Subsequently, this component generates an *entity graph* in which nodes are the annotated entities and edges are weighted with the relatedness between the edge-connected entities (Jaccard Relatednesses in Figure 1).

(4) WORD2VEC (Mikolov et al., 2013) is the component that aims to enrich the document with latent information. This component is a further addition with respect the version proposed in the conference paper Ponza et al. (2017). More precisely, it takes the entities annotated by WAT and map them into their proper continuous vector representations learned via neural networks. These latent representations are further used to compute the cosine similarities between all entities that have been annotated in the document d by WAT. Technically speaking, the WORD2VEC module is constituted by two components that respectively deploy two different kinds of *latent entity representations*: Entity2Vec (Ni et al., 2016) and DeepWalk (Perozzi et al., 2014) (more details are provided in Section 3), respectively.

Feature Generation. The second module deploys the data generated by the Document Enrichment component in order to compute a rich set of features for each entity e. In order to do so, four main components are deployed (i.e. `Standard`, `Semantic`, `Word2Vec` and `Syntactic Features`) in order to map each e into its proper vector of features. A more detailed description of these components as well as the algorithms implemented to generate the features for each entity is provided below.

Entity Salience Classification. The goal of the last component is to classify entities into their class (e.g. salient vs non-salient) given the entity features computed by the previous module. For this, SWAT deploys the efficient and highly scalable *eXtreme Gradient Boosting* software library (Chen and Guestrin, 2016) (XGBoost Classifier in Figure 1) which is trained and tested as detailed in Section 4.

While the use of the third module is pretty standard, the first and second modules are more involved and constitute the main novel part of our system SWAT. Hence, the rest of this section is devoted to detail the first two modules which generate the features for each entity that has been annotated in the input document d — called `Standard`, `Syntactic` , `Semantic` and `Word2Vec` — to be used in the third and last entity salience classification module. In order to facilitate the reading and understanding of the large number of features deployed by SWAT, we report all of them in Table 1 via a grouping that highlights their novelty (vertical label in the left most column) and the software component in charge of their implementation (rightmost column). In the text below we comment only on the new features introduced by SWAT, referring for the others to the description reported in the Table.

Position-based Features: *spread* and *bucketed-freq*. These features deploy the distribution within document d of the entities occurrences in order to predict their

Table 1: Summary of the features used in SWAT.

	Name	Description	Component
Common Features	$ef(e,d), idf(e), ef\text{-}idf(e,d)$	Entity frequency (number of times WAT annotates e in d), inverse document frequency for e and their product.	`Standard`
	$position\text{-}stats_{\{s,t\}}(e,d)$	Minimum, maximum, arithmetic mean, median, standard deviation and harmonic mean of sentence- (resp. token-) positions of e in d.	`Standard`
	$mention\text{-}title(e,d)$	Presence of a *mention* of e in the title of d.	`Standard`
	$entity\text{-}title(e,d)$	Presence of e in the title of d.	`Standard`
	$is\text{-}upper(e,d)$	True if one of the mentions of e appear in d in uppercase, false otherwise.	`Standard`
CMU-GOOGLE Features	$1st\text{-}loc(e,d)$	Index of the sentence in which the first mention of e appears in d.	`Standard`
	$head\text{-}count(e,d)$	Frequency of head word of entity e in the document d.	`Syntactic`
	$mentions(e,d)$	Sum between entity frequency and co-referenced frequency of e in d.	`Syntactic`
	$headline(e,d)$	POS tag of each word of e that appears in at least one mention and also in the headline of d.	`Syntactic`
	$head\text{-}lex(e,d)$	Lower-cased head word of the first mention of e in d.	`Syntactic`
	$google\text{-}centrality(e,d)$	PageRank score of e on the entity graph generated from d, where weights are the co-occurrence probability of two entities, computed on the training set.	`Standard`
Novel Features Adopted in SWAT	$spread_{\{s,t\}}(e,d)$	Difference between the max and min sentence- (resp. token-) positions of e in d.	`Standard`
	$bucketed\text{-}freq_{\{s,t\}}(e,d)$	Vector of bucketed frequencies through sentence- (resp. token-) positions of e in d.	`Standard`
	$textrank\text{-}stats(e,d)$	Minimum, maximum, arithmetic mean, median, standard deviation and harmonic mean of TEXTRANK scores of sentences where e appears in d.	`Syntactic`
	$dep\text{-}freq(e,d)$ $dep\text{-}bucketed\text{-}freq_{\{s,t\}}(e,d)$	Frequency of e in d when it appears as dependent of the dependency relation *dep*. Vector of bucketed frequencies through sentence- (resp. token-) positions of e in d limited to the mentions where e appears as dependent with relation *dep*.	`Syntactic` `Syntactic`
	$dep\text{-}position\text{-}stats_{\{s,t\}}(e,d)$	Minimum, maximum, arithmetic mean, median, standard deviation and harmonic mean of sentence- (resp. token-) positions of e in d, where only the mentions where e appears as dependent of a dependency relation *dep* are considered.	`Syntactic`
	$dep\text{-}textrank\text{-}stats(e,d)$	Minimum, maximum, arithmetic mean, median, standard deviation and harmonic mean of TEXTRANK scores where only the sentences where e appears as dependent of a dependency relation *dep* are taken into account.	`Syntactic`
	$comm\text{-}stats(e,d)$	Minimum, maximum, arithmetic mean, median, standard deviation and harmonic mean of the *commonness* values of e in d computed by WAT.	`Semantic`
	$\rho\text{-}stats(e,d)$	Minimum, maximum, arithmetic mean, median, standard deviation and harmonic mean of the ρ-score values of e in d computed by WAT.	`Semantic`
	$rel\text{-}stats(e,d)$	Minimum, maximum, arithmetic mean, median, standard deviation and harmonic mean of the relatedness scores between e and all other entities annotated in d.	`Semantic`
	$rel\text{-}bucketed\text{-}stats_{\{s,t\}}(e,d)$	Minimum, maximum, arithmetic mean, median, standard deviation and harmonic mean of the relatedness scores between e and all other entities present in d, bucketed over document positions (both at sentence- and token-level).	`Semantic`
	$rel\text{-}centrality(e,d)$	Degree, PageRank, Betweenness, Katz, HITS, Closeness and Harmonic (Boldi and Vigna, 2014) scores of e computed on the entity graph of d.	`Semantic`
	$wiki\text{-}id(e)$	Wikipedia identifier of e, normalized via feature hashing.	`Semantic`
	$w2v(e)$	Entity2Vec and DeepWalk embedding vectors of e.	`Word2Vec`
	$w2v\text{-}stats(e,d)$	Minimum, maximum, arithmetic mean, median, standard deviation and harmonic mean of the cosine-similarity between the Entity2Vec and DeepWalk embeddings of e and the ones of the other entities annotated in the title and headline of d.	`Word2Vec`
	$w2v\text{-}cosine(e,d)$	Cosine-similarity between the Entity2Vec and DeepWalk embeddings of e and the average of the corresponding embeddings of the words present in the title and headline of d.	`Word2Vec`

salience score. So SWAT computes the position of an entity e within the document d, in terms of tokens or sentences. For token-level features (indicated with the subscript t) it is considered the index of the first token for each mention of e, normalized by the number of tokens of d; whereas for sentence-level features (indicated with the subscript s) it is considered the index of the sentences where the entity e is annotated, normalized by the number of sentences of d. These features naturally improve the $1st\text{-}loc(e, d)$ feature introduced by (Dunietz and Gillick, 2014) thus making more robust SWAT with respect to the distribution of salient entities, as shown in Section 5.4.

Summarization-based Features: *TextRank-stats*. These features exploit the score that summarization algorithms assign to sentences that contain salient information and thus possibly contain salient entities. So SWAT computes, for each entity e, some statistical measures derived from the scores assigned by TEXTRANK (Mihalcea and Tarau, 2004) to the sentences where a mention of e occurs.

Linguistic-based Features: *dep-.** These features exploit the grammatical structure, namely the dependency trees, of sentences where the entities occur. Unlike (Dunietz and Gillick, 2014), where dependency trees are used to extract only the head of a mention, SWAT combines frequency, position and summarization information with dependency relations generated by the CORENLP's dependency parser. Moreover, SWAT takes into account the mentions m of e that have at least one token dependent of a specific dependency relation, limited to those where salient entities appear more frequently in the training set: they were preposition-in, adjective modifier, possessive, noun compound modifier and subject dependency relations.

WORD2VEC-based Features: *w2v-.** This set of features aims at modeling the annotated entities and their relationships by means of proper embeddings generated via deep neural networks. Specifically, SWAT deploys both CBOW and Skip-gram models (Mikolov et al., 2013) via two well-known algorithms:

(1) Entity2Vec (Ni et al., 2016) is an extension of the original WORD2VEC that computes a unique embedding for both entities and words extracted from the textual descriptions of the Wikipedia pages.
(2) DeepWalk (Perozzi et al., 2014) is another variation of the original WORD2VEC that computes an embedding for nodes of a graph, which is here the Wikipedia graph.

SWAT uses as features the continuous vectors derived from Entity2Vec and DeepWalk, plus several other statistics computed over their cosine-similarity measure.

Relatedness-based Features: *rel-.** These features are introduced to capture how much an entity e is related to all other entities in the input document d, with the intuition that if an entity is salient then its topic should not be a *isolated* in d. SWAT uses two main groups of relatedness functions:

(1) the Jaccard relatedness described by (Piccinno and Ferragina, 2014), since its deployment in the disambiguation phase of WAT achieves the highest performance over different datasets (Usbeck et al., 2015);
(2) the cosine-similarity between the latent embeddings of the compared entities, either based on Entity2Vec or on DeepWalk.

Furthermore, these relatedness functions are used to compute other two classes of features:

(1) the ones based on several centrality algorithms – i.e. Degree, PageRank, Betweeness, Katz, HITS, Closeness and Harmonic (Boldi and Vigna, 2014) – applied over three versions of the entity graph described in Stage 1. We recall that this is a

Table 2: Fields of the SWAT's JSON request and response, respectively.

	Name	Description	Type
Request	`title`	Title of the document.	String
	`headline`	Headline of the document.	String
	`content`	Content of the document.	String
	`mentions`	Type of detected mentions. It can be `proper` (for annotating proper nouns), `common` (for annotating common nouns) or both (for annotating both proper and common nouns).	String
Response	`status`	Status of the response. It can be `ok` or `error`.	String
	`annotations`	List of annotations. Each annotation is an object with the structure described in Table 3. It is present only if `status` equals to `ok`.	List
	`info`	Information about the encountered error. It is present only if `status` equals to `error`.	String

Table 3: Fields present in each object of `annotations` field in the JSON response.

Name	Description	Type
`wiki_id`	Wikipedia ID of the annotated entity.	Integer
`description`	Brief description of the entity.	String
`salience-boolean`	1 if the entity is salient, 0 otherwise.	Integer
`salience-score`	Score of relevance of the entity.	Float
`spans`	List of pairs of integers. Each pair contains the start (included) and end (excluded) offsets at character-level of the annotated entity in the input text.	List

complete graph where nodes are entities and edges are weighted with a similarity measure between the connected entities which is estimated either with Jaccard, or with Entity2Vec or with DeepWalk.

(2) the ones based on proper statistics aggregating the relatedness scores between the entity e and other entities in d.

3.1. Graphical User Interface and Public API

Figure 2 shows a simple GUI[5] that allows using SWAT over an input document loaded via a Web interface. In addition to the GUI, it is possible to deploy SWAT through a REST-like interface[6]. The API provides results in both human and machine-readable form, by deploying a simple JSON format (see Tables 2 and 3). In order to show how the interaction with SWAT works, we offer a Python code

[5]The demo of the system is accessible at `https://swat.d4science.org`.

[6]The API is accessible at `https://sobigdata.d4science.org/web/tagme/swat-api`.

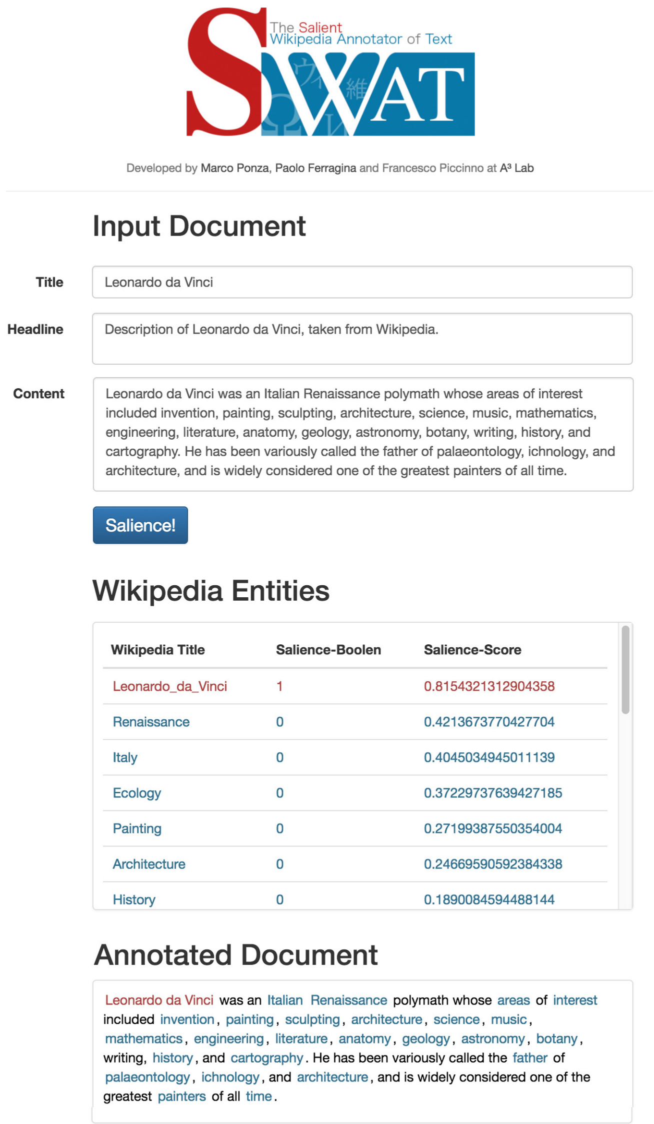

FIGURE 2: The GUI of SWAT prototype allows to detect and classify Wikipedia entities from an input text. The box `Wikipedia Entities` shows the annotated entities with a boolean label, denoting salient (red) and non-salient (blue) entities, and ranked by their `salience score`, namely XGBOOST's probability. The box `Annotated Document` shows the mentions annotated to their pertinent Wikipedia pages.

snippet in Listing 1 for querying our system and the corresponding JSON response in Listing 2. A query requires just two optional parameters (title and headline) and one mandatory parameter (the content of the document). The response includes all entities annotated by SWAT and several useful information for each of them.

Listing 1: Python code for querying the SWAT's public API. The authorization token MY_GCUBE_TOKEN is needed for using the service and obtainable through free registration.

```
import requests
url = "https://swat.d4science.org/tag"
document = { "title": "Obama travels.",
         "headline": "A toy example.",
         "content": "Barack Obama was in Pisa
         for a flying visit.",
         "mentions": "proper,common"
}
requests.post(url, document,
params={"gcube-token": MY_GCUBE_TOKEN})
```

Listing 2: JSON response after querying SWAT.

```
{   "status": "ok",
        "annotations":
        [
        {   "wiki_id": 534366,
                "wiki_title": "Barack_Obama",
                "description": "Barack Hussein Obama II
                is an American...",
                "salience-boolean": 1,
                "salience-score": 0.66,
                "spans": [ [0, 12] ]
        },
        {   "wiki_id": 24636,
                "wiki_title": "Pisa",
                "description": "Pisa is a city
                in Tuscany...",
                "salience-boolean": 0,
                "salience-score": 0.10,
                "spans": [ [20, 24] ]
        },
        {   "wiki_id": 327283,
                "wiki_title": "State_visit",
                "description": "A state visit is a
                formal visit...",
                "salience-boolean": 0,
                "salience-score": 0.10,
                "spans": [ [38, 43] ]
        }
        ]
}
```

4. VALIDATION METHODOLOGY

In this section we describe the validation methodology performed for evaluating our system SWAT. Section 4.1 outlines the metrics used in the experiments for measuring the accuracy of the systems at hand, Section 4.2 describes the datasets used in our benchmarks by reporting the main differences between the two testbeds and finally Section 4.3 describes the experimented tools whose results will be discussed in Section 5.

4.1. Evaluation Metrics

For the evaluation of the accuracy of the systems we use precision, recall and F1, as standard in Information Retrieval (Manning et al., 2008) for the assessment the quality of classification systems.

Precision. It is the ration between the predicted salient entities that are in the ground-truth with respect to all the salient entities predicted by a solution:

$$\frac{\{\texttt{correct salient entities}\} \cap \{\texttt{predicted salient entities}\}}{\{\texttt{predicted salient entities}\}} \tag{1}$$

Recall. The ratio of the salient entities of the ground-truth that have been also predicted as salient by a solution:

$$\frac{\{\texttt{correct salient entities}\} \cap \{\texttt{predicted salient entities}\}}{\{\texttt{correct salient entities}\}} \tag{2}$$

F1. It is the harmonic mean between *precision* and *recall*:

$$\frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3}$$

where $\{\texttt{correct salient entities}\}$ are the salient entities that are in the ground-truth, where $\{\texttt{predicted salient entities}\}$ are the salient entities predicted by a system. In our experiments we report both micro and macro scores of these metrics, since they have been respectively used by (Dunietz and Gillick, 2014) and (Trani et al., 2017) for validating their system. More precisely, micro scores focus their evaluation of the overall quality of the salient predictions in a dataset, whereas macro scores are computed as the average of micro metrics calculated for every single document.

4.2. Datasets

Our experimental validation of the accuracy and efficiency performance of SWAT is executed on the following datasets.

New York Times. The annotated version of this dataset, suitable for the entity salience problem, was introduced by (Dunietz and Gillick, 2014). It consists of annotated news drawn from 20 years of the New York Times newspaper (see also (Sandhaus, 2008)). It is worth to point out that the numbers reported by (Dunietz and Gillick, 2014) are slightly different from the ones we derived by downloading this dataset: authors informed us that this is due to the way they have exported annotations in the final release and this impacts onto the F1-performance of their system for about -0.5% in absolute micro-F1. We will take these figures into account in the rest of this section when comparing SWAT with the CMU-GOOGLE system.
Since the entity linker used by (Dunietz and Gillick, 2014) is not publicly avail-

able (and this was used to derive the ground truth of the NYT dataset), we kept only those entities which have been generated by SWAT and CMU-GOOGLE. The final figures are the following: the news in the training+validation set are $99,348 = 79,462 + 19,886$, and are $9,577$ in the test set; these news contain a total of $1,276,742$ entities in the training+validation set (i.e. $1,021,952 + 254,790$) and $19,714$ entities in the test set. Overall the dataset contains 108,925 news, with an average number of 975 tokens per news, more than 3 million mentions and $1,396,456$ entities, of which 14.7% are labeled as salient.

Wikinews. This dataset was introduced by (Trani et al., 2017) and consists of a sample of news published by Wikinews from November 2004 to June 2014 and annotated with Wikipedia entities by the Wikinews community. This dataset is significantly smaller than NYT in all means: number of documents (365 news), their lengths (an average of 297 tokens per document) and number of annotations (a total of $4,747$ manual annotated entities, of which 10% are labeled as salient). Nevertheless, this dataset has some remarkable features with respect to NYT: the ground-truth generation of the salient entities was obtained via human-assigned scores rather than being derived in a rule-based way, and it includes both proper nouns (as in NYT) and common nouns (unlike NYT) as salient entities. For the cleaning of the dataset we follow (Trani et al., 2017) as done in their experimental setup by removing the 61 documents that do not have any salient entity,

As far as the dataset subdivision and evaluation process are concerned, we used the following methodology. For the NYT, we use the same training/testing splitting as defined by (Dunietz and Gillick, 2014) as detailed above. For Wikinews we deploy the evaluation procedure described by (Trani et al., 2017), namely the averaged macro-F1 of a 5-fold cross-validation.

4.3. Tools

Baselines. We implemented four baselines. The first one is the same baseline introduced by (Dunietz and Gillick, 2014), it simply classifies an entity as salient if it appears in the first sentence of the input document. The other three baselines are new and try to investigate the individual power of some novel features adopted by SWAT. More precisely, the second baseline (called ρ-baseline) extends the previous one by adding the check whether the ρ-score (capturing entity coherence) is greater than a fixed threshold. The third (resp. fourth) baseline classifies an entity as salient if its maximum TextRank (resp. Rel-PageRank) score is greater than a fixed threshold.

Two Versions of the CMU-GOOGLE System. The original system (Dunietz and Gillick, 2014) uses a proprietary entity linker to link proper nouns to Freebase entities, and then classify them into salient and non-salient by deploying a small number of standard text-based features, mainly based on position and frequency. This system is not available to the public, so we will report in our Tables the performance figures published by (Dunietz and Gillick, 2014).

To support experiments over the new dataset Wikinews, we decided to implement our own version of the CMU-GOOGLE's system by substituting the proprietary modules with open-source tools: we used WAT as entity linker (Piccinno and Ferragina, 2014) and a state-of-the-art logistic regressor as classifier (Pedregosa et al., 2011). Our (re-)implementation achieves performance very close to the original system (see Table 5) and thus it is useful to obtain a fair comparison over the Wikinews dataset.

The SEL System. This is the system proposed by (Trani et al., 2017) that uses a

Table 4: Candidate values and the best configuration found by the grid-search procedure for the tuning of XGBOOST's hyper-parameters on New York Times and Wikinews datasets.

Hyper-parameters	Candidate Values	New York Times	Wikinews
`max_depth`	$\{2, 4, 6, 8\}$	8	2
`min_child_weight`	$\{6, 8, 10\}$	6	6
`gamma`	$\{0.1, 0.3, 0.5\}$	0.1	0.5
`reg_alpha`	$\{0.001, 0.01, 0.05\}$	0.001	0.05
`scale_pos_weight`	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$	2	8

machine learning regressor to detect salient entities via a set of features that is wider than the ones used in CMU-GOOGLE. This system is not available to the public, so we will report in our Tables the performance figures published by (Trani et al., 2017).

Configurations of SWAT and Baselines. We experimented upon several configuration settings of SWAT and of the baselines above, according with the characteristics of the ground-truth datasets. For NYT, where the ground-truth was generated by assuming that salient entities can be mentioned in the text only as proper nouns, we configured these systems to annotate only proper nouns detected by CORENLP; whereas for Wikinews, where the ground truth comes with no assumptions, we tested two variants: one detecting only proper nouns, and the other detecting both proper and common nouns. For the tuning of XGBOOST's classifier we performed a grid-search over typical values of its hyper-parameters, finding the *best values* (i.e. the ones performing better on the validation sets of New York Times and Wikinews) reported in Table 4.

5. ANALYSIS AND DISCUSSION

We first experiment our proposed tools SWAT against the state-of-the-art over the two datasets New York Times and Wikinews (Section 5.1). Then, we analyze and discuss several aspects of our proposed systems focusing on: (i) the generalization ability of the tested systems as a function of the used training data (Section 5.2), (ii) the dependence between the size of the training-set and the accuracy of the system (Section 5.3), (iii) the impact that features have on the quality of the predictions (Section 5.4), (iv) the time efficiency of the system according to its main components and its overall speed-up when only the most relevant features are used (Section 5.5), (v) the dependence of top-systems on the position of the salient entities within the input document, and (vi) an analysis of the limitations of the current systems in terms of the types of erroneous predictions (Section 5.7).

5.1. Experimental Results

Experimental figures on the two datasets are reported in Tables 5–6, where we denote by CMU-GOOGLE-ours our implementation of the system by (Dunietz and Gillick, 2014). This system is only slightly worse than the original one, which could depend on the differences in the NYT dataset commented above and in the deployment of open-source modules rather the Google's proprietary ones. The final performance of CMU-GOOGLE-ours is very close to what claimed by (Dunietz and Gillick, 2014), thus we decide to use this software also on the Wikinews dataset. We notice that both TextRank and Rel-PageRank baselines obtain low micro- and macro-F1 performance over both datasets. This is probably due to the characteristics

Table 5: Performance of the tested systems on the New York Times' dataset.

System	Micro			Macro		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Positional Baseline	59.1	38.6	46.7	39.0	32.7	33.0
Positional-ρ Baseline	61.9	36.9	46.2	38.5	31.0	32.0
TextRank	27.0	58.8	37.0	30.0	48.6	33.4
Rel-PageRank	20.3	62.5	30.6	21.3	**55.3**	28.0
CMU-GOOGLE	60.5	63.5	62.0	-	-	-
CMU-GOOGLE-ours	58.8	62.6	60.7	47.6	50.5	46.1
SWAT	**62.4**	**66.0**	**64.1**	**50.7**	53.6	**49.4**

of these datasets: the salient information in news is typically confined to initial positions, so those systems are drastically penalized by ignoring positional information. This statement is further supported by the results of Positional and Positional-ρ baselines: they are trivial but generally achieve better performance.

Table 5 reports the results for the experiments on the New York Times dataset. We notice that the new features adopted by SWAT allow it to outperform CMU-GOOGLE-ours by 3.4% and 3.3% over micro- and macro-F1, respectively, and CMU-GOOGLE by 2.1% in micro-F1 (macro-F1 was not evaluated by (Dunietz and Gillick, 2014)). This contribution is particularly significant because of the size of NYT.

Table 6 reports the results on Wikinews dataset. We notice that, on this dataset, the improvement achieved by SWAT against the state-of-the-art is even more significant than on NYT. In fact SWAT improves the micro-F1 of XX% with respect to CMU-GOOGLE-ours and the macro-F1 of YY% with respect to SEL.

5.2. Generalization Ability of SWAT Trained on NYT

The second question we experimentally investigated is about the *generalization ability* of the feature set used by SWAT varying the datasets over which the training and tuning phases are executed. In particular, we experimented two different configurations of our system. SWAT-CLF is SWAT trained over NYT and directly used over Wikinews; and SWAT-REG s SWAT trained over NYT but whose regressor is tuned over Wikinews maximizing the macro-F1 over the training folds.

According with Table 7, SWAT-CSF obtains performance lower than the systems specifically trained over Wikinews (as expected, see Table 6), such as SWAT and SEL, but it turns actually to be slightly better than CMU-GOOGLE-ours by +0.7% in macro-F1.

On the other hand, the tuning on Wikinews by SWAT-REG allows achieving better performance in macro-F1 than both CMU-GOOGLE-ours and SEL, of 8.7% in micro-F1 with respect to CMU-GOOGLE-ours and of 8.3% and 2.3% in macro-F1 with respect to CMU-GOOGLE-ours and SEL. These figures show that the features introduced by SWAT are flexible enough to work independently from the news source and without overfitting the large single-source training data (i.e. NYT).

5.3. Accuracy versus Training Size

We analyze the performance of the two versions of SWAT with respect to different sizes of the training data. We focus these experiments on the largest dataset available, namely New York Times.

Figure 3 provides a side-by-side comparison of the performance of the two sys-

Table 6: Performance on the Wikinews dataset. For each system we report the score obtained by the system configured to annotate either only proper nouns (top) or both proper and common nouns (down).

System	Micro			Macro		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Positional Baseline	23.3	67.0	35.0	25.2	67.0	34.0
	14.4	72.0	24.0	16.1	**72.7**	25.0
Positional-ρ Baseline	36.8	60.3	45.7	38.3	61.6	43.5
	34.1	58.5	43.1	36.2	61.3	41.9
TextRank	12.2	47.5	19.4	14.1	49.3	20.2
	5.7	49.2	10.1	6.3	50.9	10.6
Rel-PageRank	10.0	51.0	16.8	10.1	51.2	15.9
	10.6	35.8	16.4	11.1	34.8	14.7
CMU-GOOGLE-ours	41.0	60.0	49.0	42.3	61.0	46.0
	41.0	56.0	47.0	41.0	58.0	45.0
SEL	-	-	-	**61.0**	50.0	52.0
SWAT	**58.0**	64.9	**61.2**	57.7	67.0	**58.3**
	51.0	67.4	58.0	53.7	69.7	56.6

Table 7: Generalization ability of SWAT trained on NYT and tested on Wikinews. For each system we report the score obtained by the system configured to annotate either only proper nouns (top) or both proper and common nouns (down).

System	Micro			Macro		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
SWAT-CLF	35.0	72.0	47.1	37.9	73.7	46.7
	27.3	**75.7**	40.1	31.3	**78.0**	41.5
SWAT-REG	**55.9**	59.9	**57.7**	**54.0**	62.4	**54.3**
	49.3	63.1	55.1	50.6	65.9	53.3

tems when 5%, 25%, 50%, 75% and 100% of the whole training data is used. The original validation set is kept for the tuning of the hyper-parameters, as described in Section 4. Micro-precision, -recall and -F1 are finally calculated over the test-set.

The precision of SWAT increases until when 50% of the whole training size is used and by reaching a peak of 63.5. Unfortunately, when the training size increases the precision degrades by eventually losing -1.1 in performance. This degradation is due to the increase of the recall which also allows to consistently improve the micro-F1 until the whole training data is used.

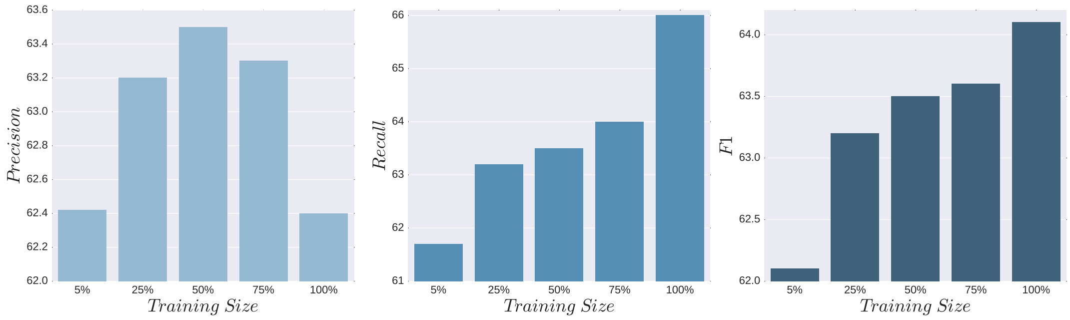

FIGURE 3: Comparison of the performance SWAT over different training sizes of the New York Times dataset.

5.4. Feature Analysis

Let us jointly discuss the most important signals emerging from the incremental feature additions experimented with SWAT on both datasets (see Figure 4). Through this analysis we aim to clarify what are the key elements needed for the entity salience detection.

Important Features for SWAT 1.0. We notice that the most important features for this system depend on four common elements: (i) *position* (e.g. title or the beginning of the document), (ii) frequency (e.g. *head-count*), (iii) the "quality" of the entity linking (e.g. ρ- and *commonness*-based features) and (iv) the relationships between entities (*jaccard-relatedness*-based features). On the other hand, several other features based on TEXTRANK and Dependency Relations appear between 20-50 and 10-20 positions for NYT and Wikinews, respectively. In spite of their marginal role, they allow SWAT to refine its predictions and eventually get the top performance claimed in Tables 5–6.

Important Features for SWAT 2.0. Complementary to the most important features of SWAT 1.0, SWAT 2.0 extends them with several novel elements that come from the WORD2VEC module. Specifically, the embedding vectors (*e2v-sg-vec*) of an entity and its cosine similarity with the words in the title (*e2v-sg-cos*) obtain a high feature score over both datasets.

5.5. Time Efficiency

The average computation time of each module constituting SWAT is reported in Figure 5. When all features are used, the most expensive component is clearly the Feature Generation module, which takes about the 64% of the whole computation time of SWAT; whereas CORENLP, WAT, TEXTRANK, WORD2VEC and CLASSIFICATION take respectively the 7%, 22%, 0.1%, 5.7% and 2% of the computation time of the whole pipeline. Conversely, when only the top-40 features learned over NYT are used, SWAT becomes much faster (up to 5x, see Figure 6) without any significant degradation on its accuracy (see Figure 4). The choice of training SWAT over NYT data is motivated by the fact that: (1) the most important features are very similar to the ones derived when the system is trained on Wikinews, and (2) the system trained on NYT and then tested on Wikinews still obtains higher performance than current state-of-the-art systems (see Section 4).

FIGURE 4: Performance of the incremental feature addition (top) of SWAT according to the corresponding feature importance provided by XGBOOST (bottom) over NYT (left) and Wikinews (right) datasets.

5.6. Flexibility over Entities' Position

In this section we address a question posed by (Dunietz and Gillick, 2014) and concerning with the evaluation of how the performance of top-systems depends on the distribution of the salient entities in the input documents. Figures 7–8 motivate further this question because they show the distribution of the salient and non-salient entities within the NYT and Wikinews datasets. As expected, most of the salient entities are concentrated on the beginning (i.e. position in the first 20%) of the news over both datasets. Moreover, the whole NYT corpus contains a significant number of them which are mentioned for the first time after the beginning of the document, with 44192 salient entities whose first position is after the first 20% of the news for a total of 31128 such news (out of the total 108925 news in NYT). On the other hand, the salient entities present in Wikinews are mainly confined at the beginning of documents, with only 28 salient entities whose first position is after the first 20% of the news. For this reason we only consider NYT as the main testbed for estimating the flexibility of the systems over entities' position, both for its large size and for the wider distribution that salient entities have inside this corpus.

Figure 9 shows the comparison among the available systems. Performance are computed only over the test set of the NYT, which contains 3911 salient entities whose first position is after the first 20% of the news, with a total of 2821 such news (which are 9577 in total in the test set). All systems are highly effective on the classification of salient entities mentioned at the beginning of the document, but their behavior differs significantly when salient entities are mentioned at the

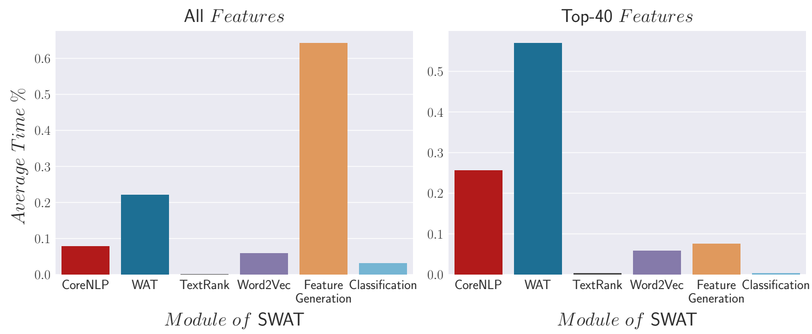

FIGURE 5: Average computation time (percentage) of each SWAT module for the whole salience-annotation pipeline by deploying all (left) and the top-40 (right) features. Performance are averaged over a sample of 400 documents of the NYT dataset.

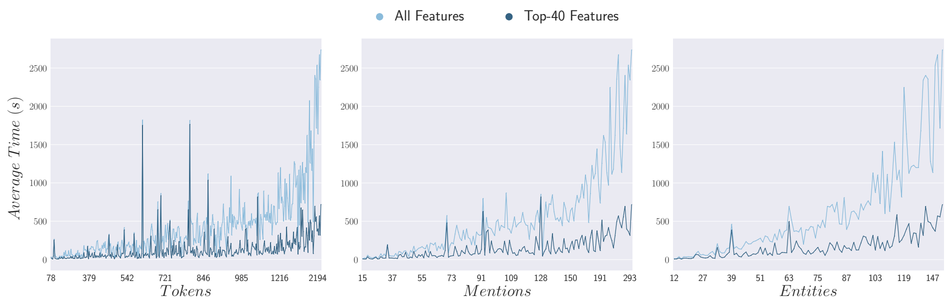

FIGURE 6: Average computation time of SWAT by distinguishing between the number of tokens, mentions and entities over a sample of 400 documents of the NYT dataset.

documents' end. In this latter case, SWAT does not overfit upon the positional feature and, indeed, obtain a high improvement with respect to CMU-GOOGLE-ours which is respectively up to 14% in micro-F1. As a consequence we can state that SWAT is more flexible with respect to salient-entities' position than CMU-GOOGLE, so that it could be used consistently over other kinds of documents where salient information is not necessarily confined to their beginning.

5.7. Error Analysis

In order to gain some insights on SWAT performance and possible improvements, we analyzed its erroneous predictions by drawing a subset of 80 (=40+40) documents from the NYT and Wikinews datasets. The most significant result we gain is what argued by Hasan and Ng (2014): namely that the deployment of semantic knowledge (i.e. Wikipedia entities) eliminates some errors that originally afflicted keyphrase extraction algorithms. However, our error analysis of 80 documents also showed that false-negative errors (i.e. entities classified as non-salient, despite being salient) are mainly due to the position-based features which frequently induce to miss a salient entity because it is not at the beginning of the news. But, on the other hand, we also noticed that a large percentage of the analyzed news of NYT ($\sim$ 35%) and Wikinews ($\sim$ 40%) contain false-positive errors which are ground-truth errors:

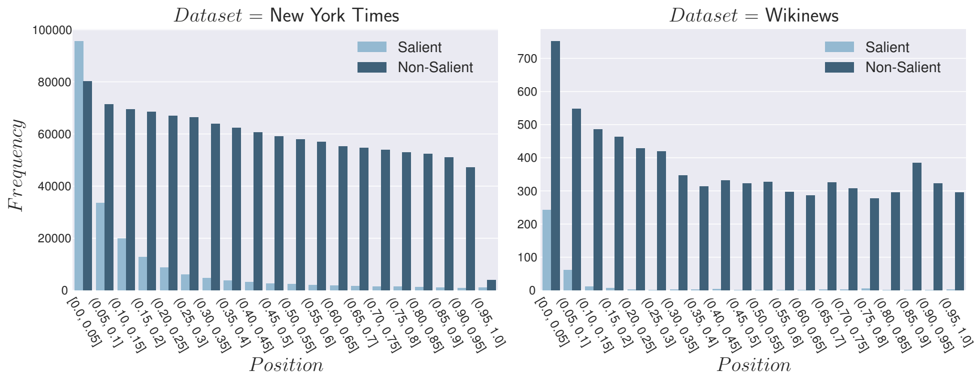

FIGURE 7: The histograms plot the frequency distribution of salient versus non-salient entities according to their *first* positions in the documents over NYT (left) and Wikinews datasets (right).

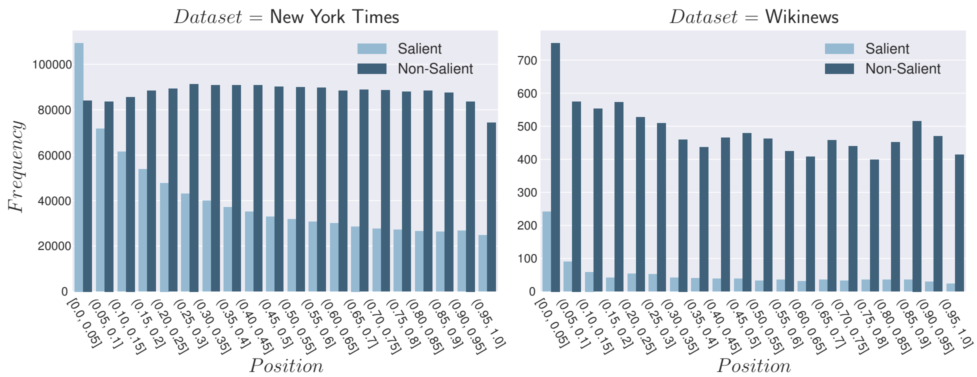

FIGURE 8: The histograms plot the frequency distribution of salient versus non-salient entities according to *all* their occurrences in the documents over NYT (left) and Wikinews datasets (right).

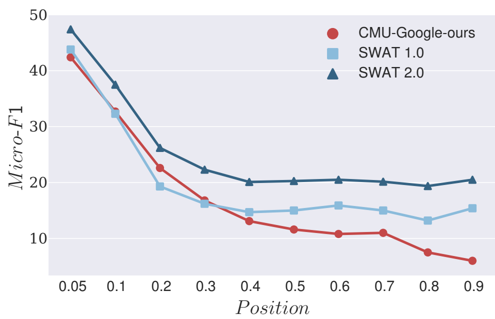

FIGURE 9: Micro-F1 performance as a function of the first token positions on the NYT dataset. Each point (x, y) indicates that the micro-F1 is y for all entities whose position is larger than x.

in these cases SWAT correctly identifies the salience of an entity, but the ground truth does not label it as salient and so it is unfortunately counted as an error in our tables.

This analysis suggests that SWAT performance could be actually higher than what we claimed before and a better ground-truth dataset should be devised, as we foresee in the concluding section.

6. CONCLUSION AND FUTURE WORK

For the contributions of this paper we refer the reader to the introduction, here we comment on four research directions that spurred from the extensive experiments we have conducted over the NYT and Wikinews datasets. The first one concerns with the improvement in the quality of the NYT dataset (which is the largest one available) by (i) augmenting its annotations with common nouns and (ii) by labeling its ground-truth via a crowdsourcing task rather than a rule-based approach as the one adopted by (Dunietz and Gillick, 2014). The second research direction concerns with the design of faster entity l,inkers which are crucial to allow the processing of large datasets, such as NYT, in reasonable time (in fact, the current annotation of NYT by WAT run on multiple threads took about 20 days). Finally, the last two directions worth of investigation concern with analyzing the performance of our system in terms of entity ranking rather than entity classification problem, and testing it over datasets of different types such as tweets, web pages or research papers.

Acknowledgments

Part of the work of the first two authors has been supported by a *Bloomberg Data Science Research Grant (2017)*, and by the EU grant for the Research Infrastructure *"SoBigData: Social Mining & Big Data Ecosystem"* (INFRAIA-1-2014-2015, agreement #654024).

REFERENCES

ANICK, PETER. 2003. Using terminological feedback for web search refinement: a log-based study. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in Information Retrieval, ACM, pp. 88–95.

BIZER, CHRISTIAN, JENS LEHMANN, GEORGI KOBILAROV, SÖREN AUER, CHRISTIAN BECKER, RICHARD CYGANIAK, and SEBASTIAN HELLMANN. 2009. Dbpedia-a crystallization point for the web of data. Web Semantics: science, services and agents on the World Wide Web, **7**(3):154–165.

BLANCO, ROI, GIUSEPPE OTTAVIANO, and EDGAR MEIJ. 2015. Fast and space-efficient entity linking for queries. *In* Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, ACM, pp. 179–188.

BOLDI, PAOLO, and SEBASTIANO VIGNA. 2014. Axioms for centrality. Internet Mathematics, **10**(3-4):222–262.

BOLLACKER, KURT, COLIN EVANS, PRAVEEN PARITOSH, TIM STURGE, and JAMIE TAYLOR. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. *In* Proceedings of the 2008 ACM SIGMOD international conference on Management of data, AcM, pp. 1247–1250.

BOVI, CLAUDIO DELLI, LUCA TELESCA, and ROBERTO NAVIGLI. 2015. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. Transactions of the Association for Computational Linguistics, **3**:529–543.

BRUZA, PD, and THEO WC HUIBERS. 1996. A study of aboutness in information retrieval. Artificial Intelligence Review, **10**(5):381–407.

CHEN, TIANQI, and CARLOS GUESTRIN. 2016. Xgboost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 785–794.

CORNOLTI, MARCO, PAOLO FERRAGINA, MASSIMILIANO CIARAMITA, STEFAN RÜD, and HINRICH SCHÜTZE. 2016. A piggyback system for joint entity mention detection and linking in web queries. *In* Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, pp. 567–578.

DUNIETZ, JESSE, and DANIEL GILLICK. 2014. A new entity salience task with millions of training examples. Volume 14 of *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, p. 205.

FERRAGINA, PAOLO, FRANCESCO PICCINNO, and ROBERTO SANTORO. 2015. On analyzing hashtags in twitter. *In* Ninth International AAAI Conference on Web and Social Media.

FERRAGINA, PAOLO, and UGO SCAIELLA. 2012. Fast and accurate annotation of short texts with wikipedia pages. IEEE Software, **29**(1):70–75.

FETAHU, BESNIK, KATJA MARKERT, and AVISHEK ANAND. 2015. Automated news suggestions for populating wikipedia entity pages. *In* Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, pp. 323–332.

GABRILOVICH, EVGENIY, and SHAUL MARKOVITCH. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. Volume 7 of *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1606–1611.

GAMON, MICHAEL, TAE YANO, XINYING SONG, JOHNSON APACIBLE, and PATRICK PANTEL. 2013. Identifying salient entities in web pages. Proceedings of the 22nd ACM international conference on Information & Knowledge Management (CIKM), ACM, pp. 2375–2380.

HASAN, KAZI SAIDUL, and VINCENT NG. 2014. Automatic keyphrase extraction: A survey of the state of the art. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1262–1273.

LIU, XIAOHUA, YITONG LI, HAOCHENG WU, MING ZHOU, FURU WEI, and YI LU. 2013. Entity linking for tweets. *In* ACL (1), pp. 1304–1311.

LIU, ZHIYUAN, WENYI HUANG, YABIN ZHENG, and MAOSONG SUN. 2010. Automatic keyphrase extraction via topic decomposition. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 366–376.

MANNING, CHRISTOPHER D, MIHAI SURDEANU, JOHN BAUER, JENNY ROSE FINKEL, STEVEN BETHARD, and DAVID MCCLOSKY. 2014. The stanford CORENLP natural language processing toolkit. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL, System Demonstrations), pp. 55–60.

MANNING, D CHRISTOPHER, PRABHAKAR RAGHAVAN, and SCHÜTZE HINRICH. 2008. Introduction to information retrieval. An Introduction To Information Retrieval, **151**:177.

MEIJ, EDGAR, WOUTER WEERKAMP, and MAARTEN DE RIJKE. 2012. Adding semantics to microblog posts. *In* Proceedings of the fifth ACM international conference on Web search and data mining, ACM, pp. 563–572.

MIHALCEA, RADA, and PAUL TARAU. 2004. Textrank: Bringing order into text. Volume 4 of *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 404–411.

MIKOLOV, TOMAS, ILYA SUTSKEVER, KAI CHEN, GREG S CORRADO, and JEFF DEAN. 2013. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems (NIPS), pp. 3111–3119.

NAVIGLI, ROBERTO, and SIMONE PAOLO PONZETTO. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence, **193**:217–250.

NGUYEN, DAT BA, ABDALGHANI ABUJABAL, NAM KHANH TRAN, MARTIN THEOBALD, and GERHARD WEIKUM. 2017. Query-driven on-the-fly knowledge base construction. Proceedings of the VLDB Endowment, **11**(1).

NI, YUAN, QIONG KAI XU, FENG CAO, YOSI MASS, DAFNA SHEINWALD, HUI JIA ZHU, and SHAO SHENG CAO. 2016. Semantic documents relatedness using concept graph representation. Proceedings of the 9th ACM International Conference on Web Search and Data Mining (WSDM), pp. 635–644.

NIU, FENG, CE ZHANG, CHRISTOPHER RÉ, and JUDE W SHAVLIK. 2012. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. VLDS, **12**:25–28.

PARANJPE, DEEPA. 2009. Learning document aboutness from implicit user feedback and document structure. Proceedings of the 18th ACM Conference on Information and knowledge Management (CIKM), pp. 365–374.

PEDREGOSA, FABIAN, GAËL VAROQUAUX, ALEXANDRE GRAMFORT, VINCENT MICHEL, BERTRAND THIRION, OLIVIER GRISEL, MATHIEU BLONDEL, PETER PRETTENHOFER, RON WEISS, VINCENT DUBOURG, and OTHERS. 2011. Scikit-learn: Machine learning in python. Journal of Machine Learning Research, **12**(Oct):2825–2830.

PEROZZI, BRYAN, RAMI AL-RFOU, and STEVEN SKIENA. 2014. Deepwalk: Online learning of social representations. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM, pp. 701–710.

PICCINNO, FRANCESCO, and PAOLO FERRAGINA. 2014. From TagME to WAT: a new entity annotator. Proceedings of the 1st International Workshop on Entity Recognition & Disambiguation (ERD at SIGIR), ACM SIGIR, pp. 55–62.

PONZA, MARCO, PAOLO FERRAGINA, and FRANCESCO PICCINNO. 2017. Document aboutness via sophisticated syntactic and semantic features. Proceedings of the 22nd International Conference on Applications of Natural Language to Information Systems (NLDB), Springer, pp. 441–453.

RADLINSKI, FILIP, ANDREI BRODER, PETER CICCOLO, EVGENIY GABRILOVICH, VANJA JOSIFOVSKI, and LANCE RIEDEL. 2008. Optimizing relevance and revenue in ad search: a query substitution approach. Proceedings of the 31st International ACM SIGIR conference on Research and development in Information Retrieval (SIGIR), pp. 403–410.

SANDHAUS, EVAN. 2008. The new york times annotated corpus. Linguistic Data Consortium, Philadelphia, **6**(12):e26752.

SCAIELLA, UGO, PAOLO FERRAGINA, ANDREA MARINO, and MASSIMILIANO CIARAMITA. 2012. Topical clustering of search results. Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM), ACM, pp. 223–232.

SCHUHMACHER, MICHAEL, LAURA DIETZ, and SIMONE PAOLO PONZETTO. 2015. Ranking entities for web queries through text and knowledge. *In* Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, pp. 1461–1470.

SUCHANEK, FABIAN M, GJERGJI KASNECI, and GERHARD WEIKUM. 2007. Yago: a core of semantic knowledge. Proceedings of the 16th international conference on World Wide Web (WWW), pp. 697–706.

TRANI, SALVATORE, CLAUDIO LUCCHESE, RAFFAELE PEREGO, DAVID E LOSADA, DIEGO CECCARELLI, and SALVATORE ORLANDO. 2017. Sel: A unified algorithm for salient entity linking. *In* Computational Intelligence.

TURNEY, PETER D. 2000. Learning algorithms for keyphrase extraction. Information Retrieval, **2**(4):303–336.

USBECK, RICARDO, MICHAEL RÖDER, AXEL-CYRILLE NGONGA NGOMO, CIRO BARON, ANDREAS BOTH, MARTIN BRÜMMER, DIEGO CECCARELLI, MARCO CORNOLTI, DIDIER CHERIX, BERND EICKMANN, and OTHERS. 2015. Gerbil: general entity annotator benchmarking framework. Proceedings of the 24th International Conference on World Wide Web (WWW), pp. 1133–1143.

VITALE, DANIELE, PAOLO FERRAGINA, and UGO SCAIELLA. 2012. Classification of short texts by deploying topical annotations. *In* Proceedings of the 34th European Conference on IR Research (ECIR, 2012), pp. 376–387.

VRANDEČIĆ, DENNY, and MARKUS KRÖTZSCH. 2014. Wikidata: a free collaborative knowledgebase. Communications of the ACM, **57**(10):78–85.