An Assessment of Online Semantic Annotators for the Keyword Extraction Task

Ludovic Jean-Louis[1], Amal Zouaq[1,2], Michel Gagnon[1], and Faezeh Ensan[2]

[1] Ecole Poytechnique de Montreal, Montreal, Canada
{ludovic.jean-louis,michel.gagnon}@polymtl.ca
[2] Royal Military College of Canada, Kingston, Canada
amal.zouaq@rmc.ca, faezeh.ensan@gmail.com

Abstract. The task of keyword extraction aims at capturing expressions (or entities) that best represent the main topics of a document. Given the rapid adoption of these online semantic annotators and their contribution to the growth of the Semantic Web, one important task is to assess their quality. This article presents an evaluation of the quality and stability of semantic annotators on domain-specific and open domain corpora. We evaluate five semantic annotators and compare them to two state-of-the-art keyword extractors, namely KP-miner and Maui. Our evaluation demonstrates that semantic annotators are not able to outperform keyword extractors and that annotators perform best on domains having a high keyword density.

1 Introduction

The task of keyword extraction aims at capturing expressions (or entities) that best represent the main topics of a document. Keywords are widely used in text processing applications for different purposes. Typical tasks that rely on keyword extraction include: producing a list of the key-phrases from document content [1; 2; 3], filtering documents [4] and suggesting additional resources (e.g. advertisements) based on the keywords in document content [5; 6; 7]. Another important application of keyword extraction is semantic annotation, which involves keyword spotting and disambiguation. In the past few years, various online RESTFul APIs have been made available for analysing documents and enriching their content with semantic annotations. Usually the APIs (or extractors) provide one or several services for various tasks including keyword extraction, named entity extraction and concept (sometimes called *topic*) extraction. Besides spotting relevant entities in text, some services provide contextual disambiguation of the entities: they identify a concept in a given knowledge base that corresponds to a text fragment. Since Wikipedia has a large coverage of different domains and entities, it is often used by these APIs as a disambiguation knowledge base [8; 9].

Given the rapid adoption of online semantic annotators and their contribution to the growth of the Semantic Web, one important task is to assess their quality. Our objective in this study is to measure the performances of the semantic annotators and keyword extractors when considering domain-specific and open domain documents. Precisely, we compare the systems on two datasets, the SemEval 2010 evaluation corpus [10; 11] that we consider as domain-specific (computer science) and the newswire corpus

Crowd500 [12] that can be considered both as open domain and domain-specific. In fact, Crowd500 is composed of documents from diverse domains, hence we view it as an "open domain" corpus composed of several domain-specific sub-corpora. Both datasets are presented in Section 4.1.

As previously said, one objective in this study is to compare the performance of semantic annotators to state-of-the art keyword extractors. In fact, it seems natural to expect that the spotting task in semantic annotators reaches the same performance as keyword extractors. In particular, we consider two available systems among the top participants in the SemEval 2010 competition, KP-Miner [3] and Maui[3] [13]. These systems have demonstrated a quite good performance on the domain-specific SemEval corpus, and we are interested in comparing them to semantic annotators results, and evaluating their performance in an open-domain context. All the systems that are considered in this experiment are presented in Section 3.

The rest of the paper is organized as follows: Section 2 covers previous works on keyword extraction as well as previous assessments of text annotation services. In Section 3, we briefly introduce each individual keyword extractor and the semantic annotators. A detailed assessment of the extractors results is presented in Section 4, followed by a discussion in Section 5 and a conclusion (Section 6).

2 Related work

Usually, keyword extraction algorithms rely on three steps for extracting keywords: i) candidate keyword extraction also referred to as *candidate keyword generation*; ii) candidate keywords ranking; iii) keyword selection. The *extraction* stage aims at selecting phrases from the document that are potential keywords. The *ranking* stage associates a confidence score to each candidate and sorts them (usually in decreasing order) according to that score. Finally, the *selection* stage collects a subset of k elements from the highly ranked keywords. Note that the *ranking* and *selection* stages are sometimes combined.

The initial step, *candidate keyword generation*, consists in enumerating all the n-grams up to a certain size in a document. Following this process, the list of n-grams can be refined by discarding the n-grams that either start or end with a stopword. Linguistic methods can also be used to refine the n-grams as presented in [14].

In the *extraction* step, which is at the heart of most keyword extraction systems, various methods have been proposed. Statistical-based methods such as KP-Miner [3] and [1; 2] are domain independent and often rely on statistical metrics such as the frequency of terms in documents. Machine learning methods such as the one employed in Maui[15; 16; 13] usually rely on supervised algorithms and therefore require an annotated corpus to train their model. In this type of approach, the *extraction* stage is viewed as a classification problem where the goal is to determine whether an n-gram is a valid keyword. Various machine learning algorithms are employed for that task. For example, Maui is based on bagged decision trees, [5] uses a maximum entropy model, and [17] relies on conditional random fields. It is difficult to conclude which model is

[3] `http://code.google.com/p/Maui-indexer/`

the most suitable for the *extraction* stage as the features used to train a given model are also important. During the SemEval evaluation, the bagged decision trees model seemed to achieve higher performance than other models. The other approaches used for the *extraction* stage include linguistic-based approaches [18] and graph-based methods [19; 20; 21; 22]. In [22], the authors compare different centrality measures and analyze their performance on different datasets including the SemEval 2010 corpus. They show that centrality-based measures outperform the TF-IDF baseline. However the authors do not compare these measures against other SemEval participants that achieve higher results. In this latter case, centrality-based metrics are largely outperformed. Last but not least, in the past few years, different works have leveraged the structured knowledge from Wikipedia to extract keywords [23; 24]. For instance, [23] exploit the titles of Wikipedia articles as well of the graph of Wikipedia categories, and [24] utilize Wikipedia as a thesaurus for candidate keyword selection.

Due to the relative new development of semantic annotators, few independent studies have been undertaken to evaluate the quality of the annotations. Previous works related to the assessment of text annotators include [25; 26; 27; 28; 29; 30; 31]. Works such as [25; 26; 27; 28; 31] mainly tackled named entity detection and entity disambiguation (associating a textual expression to an entity in a knowledge base without considering keyword/topic identification). In a similar perspective, [30] proposed an assessment of the accuracy of REST-based annotators in identifying domain-relevant named entities and topics, however their work did not compare semantic annotators with keyword extractors. [29] also has some similarity with our work. Precisely, the authors presented an analysis of two approaches for producing *tags* (identical to our keywords) to label the content of scientific articles. They presented two methods based on dictionary matching using noun phrases extracted from a corpus (ArXiv articles) and Wikipedia.

Our work differs from these previous works in several ways. In [29], the authors experimented on a single corpus. Here we compare the extractors against domain-specific and open domain corpora. Another important feature of our study is that we compare keyword extractors against the semantic annotators for the task of keyword extraction. [27; 28] focused on evaluating different systems for the named entity recognition and disambiguation tasks. Our main objective is different: instead of focusing on the named entities alone, we consider phrases that can be used to produce a list of the main topics in a document.

3 Overview of keyword extractors and semantic annotators

In this assessment, we consider several available RESTFul APIs that provide text annotation services and compare them with keyword extractors. Named entities are generally restricted to some predefined types (person, date or location) that do not always focus on the main information contained in a document. Therefore, semantic annotators that process only named entities were discarded. Table 1 mentions the keyword extractors and semantic annotators included in this experiment together with those evaluated in previous works [27; 28]. More generally, systems selection was driven by the fact that keyword extractors and semantic annotators had to provide a score (either a relevance

or confidence score) that reflects the pertinence of the keywords for the document. Note that we also considered DBPedia Spotlight[4], one commonly cited semantic annotator, as part of our initial system selection. DBPedia Spotlight provides a service for mining keywords through a dedicated keyword spotter. However, this keyword spotter returns all the n-grams contained in a document without any ranking and therefore this system was discarded in our study.

Table 1: List of the evaluated keyword extractors and semantic annotators.

Extractor/Annotator	[27]	[28]	Our work
AIDA	✓		
AlchemyAPI		✓	✓
DBpedia Spotlight	✓	✓	
Extractiv		✓	
Illinois Wikifer	✓		
KP-Miner			✓
Lupedia		✓	
Maui			✓
OpenCalais		✓	✓
Saplo		✓	
TagMe	✓		✓
TextRazor		✓	✓
Wikimeta		✓	
Wikipedia-miner	✓		
Yahoo! Content Analysis		✓	
Zementa			✓

3.1 SemEval keyword extractors

We studied several systems from the SemEval competition for comparison purposes. Our choice was constrained by the availability of these systems. Only Maui and KP-Miner were made available by the SemEval participants (Section 3.1).

KP-Miner [3] is intended for domain independent keyword extraction and relies on three steps for identifying keywords. First, some heuristics based on terms frequencies and positions are used to identify potential keywords. Then a score is given to each candidate based on a weighted version of TF-IDF score (a boost factor is introduced to favor compound terms instead of single terms). Finally a refinement process is applied. The goal is to re-order the final list of keywords by taking into account the overlap between long keywords and short ones: the weight of a keyword is decreased when a sub-part of this keyword is found in another candidate.

[4] `https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki`

Maui is the successor of the KEA system [15] and extends the feature set used by KEA with several Wikipedia-based features: i) *Wikipedia keyphraseness:* likelihood of a term in being a link in Wikipedia; ii) *semantic relatedness:* semantic score derived from Wikipedia associated with each candidate keyword; and iii) *inverse Wikipedia linkage*, which is the normalized number of pages that link to a Wikipedia page. The *semantic relatedness* feature is calculated by linking each candidate keyword to a Wikipedia article, then comparing – by means of a Wikipedia-based semantic measure – a given candidate keyword to all the other candidates. The final value corresponds to the total of all the pairwise comparisons. Maui is based on the supervised algorithm Meta-Bagging with decision trees. The system was trained on the SemEval training dataset. We relied on the model provided on the Maui user support group Web site[5] without any tuning or modification.

3.2 Online semantic annotators

We experimented five online semantic annotators, namely AlchemyAPI[6], Zemanta[7], OpenCalais[8], TagMe[9], and TextRazor[10].

AlchemyAPI is a commercial system that provides text annotation services such as entity extraction, sentiment analysis and text categorization. AlchemyAPI provides a RESTFul API with a limited number of free calls per day. The keyword extraction service returns a relevance score with each keyword. [27] and [28] considered the entity extraction service for their experiments. Here we focus on the keyword extraction (*Alch_Key*) and concept tagging (*Alch_Con*) services.

Zemanta is a commercial system that have been integrated to various recommendation systems to suggest significant terms associated with web pages or links to images. Zemanta uses the *DMOZ*[11] classification to provide keywords that are not necessarily part of the initial document. When processing a document, Zemanta adds various annotations to the document, links to related web pages or images, markups, etc. For this study we only consider the keywords service (*Zem_Key*).

OpenCalais is a commercial service that performs content enrichment via several services (relation extraction, entity disambiguation, etc.). Here we focus on the *SocialTags* (*Calais_Soc*) service. The *SocialTags* service tries to emulate how a person would tag a document using common knowledge.

TagMe is a system developed to identify and link meaningful entities in a document [32]. TagMe is particularly suited for short texts but it can also process large documents. Here we only focus on the *Spotting* service which identifies relevant text fragments in a document without disambiguation.

[5] `https://code.google.com/p/maui-indexer/downloads/detail?name=keyphrextr.tar.gz&can=2&q=`
[6] `http://www.alchemyapi.com/api/keyword-extraction/`
[7] `http://developer.zemanta.com/`
[8] `http://www.opencalais.com/`
[9] `http://TagMe.di.unipi.it/`
[10] `http://www.textrazor.com/`
[11] `http://www.dmoz.org/`

TextRazor is a commercial service comprised of different modules for text extraction. Here we focus on the *Topic Tagging* module that leverages Wikipedia to label the topics in a document. Two types of topics are annotated: high level topics or CoarseTopics (*TxtRaz_Coa*) and regular topics (*TxtRaz_Top*). For the regular topics, an ensemble of machine learning techniques are utilized to help disambiguate and link entities in the document. The models are trained based on Wikipedia articles and news stories. The high level topics are extracted using Wikipedia links and category graph and various graph centrality measures to identify the most relevant topics given the document content. For this study, we discarded the high level topics as they are generally not part of the initial document.

We experimented all the systems with their initial configuration without changing any parameter. When it was possible or required to specify the number of results to be returned by a system, we considered 50 values. Otherwise, if this parameter was not mandatory, we kept the default value.

4 Evaluation

In this section, we briefly introduce the SemEval and Crowd500 corpora before presenting the individual performance of the systems and analyzing the quality of their extraction.

4.1 Corpus description

The SemEval [10; 11] dataset is a standard benchmark in the keyword extraction field. It is comprised of 244 scientific articles, usually composed of 6 to 8 pages. The articles cover different research areas of the ACM classification related to Computer Science: Computer-Communication Networks (C), Information Storage and Retrieval (H), Artificial Intelligence (I) and Computer Applications (J). The annotation of the gold standard was carried out by human annotators who assigned a set of keywords to each document. Besides the annotators keywords, the gold standard also considers keywords originally assigned by the authors of the papers. On average, 75% of the keywords were provided by the annotators and 25% by the authors. During the evaluation, the corpus was divided into two parts corresponding to the training and testing stages: 144 articles (2265 keywords) were dedicated to training and the other 100 articles (1443 keywords) served for the final evaluation.

The Crowd500 dataset contains 500 news articles. These articles cover ten different domains and were annotated by human annotators using a crowdsourcing approach [12]. The list of domains and the number of documents per domains are presented in Table 2. The last column in Table 2 shows the keyword density ρ for each domain. The keyword density is defined as "the average number of keywords in a window of 100 words". ρ is the average keyword density when considering all documents of a domain. High values of ρ indicate that more keywords appear in the documents associated with a domain. Results in Table 2 show that the keyword density is much higher in the Crowd500 corpus than in the SemEval corpus. The "Word Politics" domain has the

highest keyword density value while the lowest one goes to the "Information Storage and Retrieval" domain. Note that documents in the SemEval corpus are longer than the ones in the Crowd500 corpus, and as a result, they contain more passages without keywords than the ones in Crowd500.

Table 2 shows that the training and testing sets are unbalanced on the Crowd500 corpus, where only 10% of the documents are dedicated to testing. In this study, our focus is on the evaluation of the systems, as opposed to their training. Consequently, the small size of the Crowd500 testing dataset is not problematic.

Table 2: Number of documents for each domain in the gold standard

SemEval	Train	Test	ρ
Computer-Communication Networks (C)	34	25	1.1
Information Storage and Retrieval (H)	39	25	0.8
Artificial Intelligence (I)	35	25	1.0
Computer Applications (J)	36	25	1.0
Total	144	100	-
Crowd500	**Train**	**Test**	ρ
Art and Culture (A)	45	5	14.9
Business (B)	45	5	14.6
Crime (Cr)	45	5	16.4
Fashion (F)	45	5	14.0
Health (He)	45	5	15.1
US politics (U)	45	5	16.8
World politics (W)	45	5	18.7
Science (Sc)	45	5	15.7
Sport (Sp)	45	5	13.7
Technology (T)	45	5	14.2
Total	450	50	-

4.2 Evaluation of the systems

This section presents the performance of the systems on both SemEval and Crowd500. Results are reported in Table 3 and Table 4 for the training and testing sets. The results are returned in terms of recall, precision and F1-score. For the SemEval dataset, we used both authors and readers selected keywords as a gold standard. Also, for this dataset, the output of the systems are normalized through stemming (we used the Porter Stemmer[12]). Both tables indicate the results for the top-15 keywords extracted by the systems[13] and all the keywords returned by the systems. These top-15 keywords were

[12] `http://tartarus.org/martin/PorterStemmer/`

[13] We report the results of the top-15 keywords to allow the comparison with the SemEval campaign results.

obtained based on the confidence scores returned by the systems. Note that our analysis is not influenced by the training data in the SemEval dataset as we do not train the systems for this study. Thus, it was reasonable to compute semantic annotators' results on both the training and testing datasets. However, we must note that the keyword extractors Maui and KP-Miner were trained on the SemEval training data, which explains their somewhat better performance on this dataset (see Maui results in Table 3).

Table 3: Evaluation of the extractors on SemEval and Crowd500 training corpora. Recall (R.), Precision (P.), F1-score (F.)

		SemEval			*Crowd500*		
API	*k*	*R. (%)*	*P. (%)*	*F. (%)*	*R. (%)*	*P. (%)*	*F. (%)*
Alch_Con	15	9.13	18.47	12.22	2.35	15.19	4.07
Alch_Key	15	21.55	22.18	21.86	4.86	18.03	7.66
Calais_Soc	15	8.77	12.23	10.22	0.04	5.93	0.07
KP-Miner	15	25.46	26.2	25.83	6.51	41.19	11.25
Maui	15	**40.85**	**42.04**	**41.43**	9.05	35.51	14.43
TagMe	15	6.12	6.3	6.21	**10.1**	**33.57**	**15.52**
TxtRaz_Top	15	1.62	1.67	1.64	4.75	16.24	7.35
Zem_Key	15	6.93	13.37	9.13	4.2	26.06	7.23
Alch_Con	All	9.13	18.47	12.22	2.35	15.19	4.07
Calais_Soc	All	8.77	12.23	10.22	2.72	13.69	4.54
Alch_Key	All	38.1	12.24	18.53	15.08	19.36	16.95
KP-Miner	All	39.77	12.28	18.76	13.84	39.66	20.52
Maui	All	**55.15**	**17.03**	**26.02**	20.72	29.45	24.33
TagMe	All	46.38	1.04	2.03	**35.77**	**22.81**	**27.86**
TxtRaz_Top	All	22.72	0.76	1.46	11.75	7.03	8.8
Zem_Key	All	6.93	13.37	9.13	4.2	26.06	7.23

4.3 Analysis of the output of the systems

Results in Table 3 and Table 4 indicate that all the systems are only able to achieve less than 30% F1-score for the keyword extraction task. If we consider the testing sets, the best results are achieved by TagMe on the Crowd500 corpus and by KP-Miner on the SemEval dataset. If we focus on the testing dataset and the "All" category, we can observe that results on the Crowd500 corpus are globally higher than the ones on the SemEval corpus: the average F1-score is 14.42% on Crowd500 while it is 9.53% on SemEval. This difference is mainly due to TagMe as the API performs poorly on SemEval but achieves good performance on Crowd500. This confirms that TagMe performs best on small documents as opposed to long documents (*c.f* Section 3.2).

The top performing systems on Crowd500 corpus are TagMe, Maui and KP-Miner (F1-score: 26.51%|23.34%|21.27%). The top three systems on SemEval are Alchemy Keyword, KP-Miner, Maui, (F1-score: 17.62%|17.23%|16.46%).

Table 4: Evaluation of the extractors on SemEval and Crowd500 testing corpora. Recall (R.), Precision (P.), F1-score (F.)

		SemEval			*Crowd500*		
API	*k*	*R. (%)*	*P. (%)*	*F. (%)*	*R. (%)*	*P. (%)*	*F. (%)*
Alch_Con	15	6.07	11.41	7.93	2.81	16.71	4.82
Alch_Key	15	21.35	21.08	21.21	6.32	21.63	9.78
Calais_Soc	15	6.75	8.98	7.71	0.09	6.67	0.17
KP-Miner	15	**26.47**	**25.87**	**26.16**	8.05	41.33	13.48
Maui	15	21.15	20.67	20.9	9.78	35.87	15.37
TagMe	15	4.5	4.4	4.45	**11.21**	**34.53**	**16.93**
TxtRaz_Top	15	1.84	1.8	1.82	5.02	15.78	7.62
Zem_Key	15	4.84	8.87	6.27	5.15	29.75	8.78
Alch_Con	All	6.07	11.41	7.93	2.81	16.71	4.82
Alch_Key	All	**37.79**	**11.49**	**17.62**	16.71	20.07	18.24
Calais_Soc	All	6.75	8.98	7.71	2.6	12.4	4.29
KP-Miner	All	37.99	11.14	17.23	14.46	40.19	21.27
Maui	All	36.29	10.64	16.46	20.3	27.46	23.34
TagMe	All	44.13	0.96	1.88	**35.89**	**21.02**	**26.51**
TxtRaz_Top	All	19.1	0.61	1.18	11.52	6.28	8.13
Zem_Key	All	4.84	8.87	6.27	5.15	29.75	8.78

Regarding the semantic annotators, the result show that they do not perform as well as the experimented keyword extractors. The lowest results are achieved by TextRazor Topics, Alchemy Concepts and Calais Social Tags. One reason is that these services mainly return topics that are not part of the documents while the gold standard is mainly composed of keywords that occur in the documents (in the SemEval corpus, only some of the authors' provided keywords are not part of documents). For Calais Social Tags, we can see that the results are particularly low on the Crowd500 corpus. The first explanation is that the API returns too few keywords, on average less than ten keywords/document (9.83 precisely) on this dataset while the gold standard contains on average 49 keywords/document (49.23 precisely). Another reason is that Calais Social Tags essentially returns tags that are derived from Wikipedia (either labels of Wikipedia pages or named entities) while the gold standard is comprised of generic terms.

4.4 Domain analysis

In the previous section, we focused on the overall performance of the systems at a high level, without considering the domains. This section provides a different perspective taking into account the domains related to the documents. Table 5 shows the results achieved by the different systems on all the domains described in Table 2. In addition to these domains, we built a "generic corpus" by randomly sampling 100 documents from both SemEval and Crowd500 corpora. The results for the "generic corpus" are represented by the "Random" entry in the table.

Table 5 indicates that when the different domains are isolated, TagMe outperforms the other systems on most of the domains related to the Crowd500 dataset (Health, Fash-

ion, etc.). The domains where TagMe performs poorly are the ones where Maui obtains high results (Artificial intelligence, Computer application, Computer communication, Information retrieval). Note that TagMe performs very poorly on these domains.

If we consider the median F1-score, the results show that the top four systems can be ranked as TagMe>Maui>KP-Miner>Alch_Key. From the average F1-score perspective, the ranking becomes Maui>TagMe>KP-Miner> Alch_Key. Finally, it is interesting to note that the results are different on the random domain, where the ranking is Maui>KP-Miner>Alch_Key>Zem_Key (notice however the quite big difference between Zem_Key and Alch_Key).

To evaluate the stability of each system across different domains, we computed the standard deviation based on the F1-scores (represented by line *STDEV* in Table 5). Standard deviation values close to zero indicate that a system has a high stability, namely the F1-scores are consistent when moving from one domain to the other. High standard deviation values show instability when changing domains. In our case, Zemanta (1.16%) is the most stable system while TagMe is the least stable one (12.5%). The standard deviation value for TagMe confirms our previous observations (Section 4.3) about this system.

From a global perspective, the systems with the lowest standard deviation values, Zemanta, Calais Social Tag and TextRazor Topics are also the ones that obtained the lowest overall performance (*c.f* Table 4). The fact that they have low scores on most of the domains tend to increase their stability.

Table 5: Evaluation of the systems on different domains (F1-score (%)).

Domain	Alch_Con	Alch_Key	Calais_Soc	KP-Miner	Maui	TagMe	TxtRaz_Top	Zem_Key
Random	4.76	11.08	4.62	14.12	**16.5**	3.45	1.65	5.73
Art and culture	4.16	19.23	5.82	18.83	23.7	**28.29**	4.85	7.83
Artificial intelligence	10.29	17.54	8.99	17.01	**20.01**	2.1	1.5	8.01
Business	4.37	13.83	4.04	22.63	24.89	**25.06**	5.59	6.66
Computer application	14.5	19.77	9.45	20.53	**24.01**	2.32	1.51	8.7
Computer communication	9.2	16.15	8.32	17.21	**21.21**	1.76	1.13	7.65
Crime	4.34	18.55	3.42	22.13	24.89	**29.34**	6.7	8.17
Fashion	3.52	14.11	4.08	16.83	21.93	**25.14**	5.55	6.32
Health	5.96	16.59	5.23	21.35	24.48	**26.88**	6.95	7.45
Information retrieval	8.09	18.98	9.96	17.78	**23.13**	1.75	1.26	7.58
US politics	3.43	23.5	4.85	22.12	25.68	**30.07**	5.27	8.72
World politics	4.58	21.98	3.93	25.17	25.84	**32.12**	6.23	9.23
Science	4.16	16.31	4.6	19.78	23.06	**27.09**	6.26	5.91
Sport	1.98	14.94	3.46	19.44	24.81	**24.92**	3.92	5.98
Technology	4.91	17.19	5.68	19.61	24.42	**26.95**	5.36	8.97
Median	4.48	17.37	5.04	19.70	24.22	**26.01**	5.32	7.74
Average	5.96	17.76	5.85	20.03	**23.72**	20.27	4.43	7.66
STDEV	3.28	3.21	2.27	2.82	2.48	**12.5**	2.2	1.16

When focusing on the domains, World politics, US politics and Crime are the domains where the systems were able to extract the most keywords successfully. Their average performance on these domains are respectively 16.13%, 15.46% and 14.69% F1-score. The most difficult domains for the semantic annotators are Computer communication, Artificial intelligence and Information retrieval (when excluding the Random domain). The average performance for these domains are respectively 10.33%, 10.68% and 11.07%. The previous finding on keyword density (see Table 2) is consistent with the observations related to the top-3 and bottom-3 domains: the top-3 domains have the highest keyword density scores, while the bottom-3 domains have the lowest density scores.

5 Discussion

From a general perspective, the results presented in this assessment demonstrate that the keyword extraction task remains a difficult and challenging task: the best systems are under the limit of 30% F1-score, whether we consider domain-specific or open-domain corpora.

The 30% F1-score limit was already observed during the SemEval 2010 shared task [10; 11] where the best system achieved 27.5% F1-score. Our results show that semantic annotators generally perform poorly on this dataset: except for Alchemy (Alch_Key), no other system was able to achieve results comparable to state of the art keyword extractors such as KP-Miner[14] or Maui. Another interesting fact is that semantic annotators are generally not able to outperform the SemEval baseline which simply selects keywords by computing TF-IDF scores for n-grams [10; 11]. The SemEval baseline achieved 15.1% F1-score (for the top-15 keywords) while most of the semantic annotators achieved less than 10%.

In [30], the authors used semantic annotators for the task of extracting domain relevant expressions. Compared to this work, we use a larger dataset and we also evaluate keyword extractors as a baseline for comparison. Among the systems experimented in [30], the best performance was achieved by Alchemy. While our results confirm the good performance of Alchemy on the SemEval corpus, they also indicate that other systems (KP-Miner, Maui and TagMe) could provide better results for the keyword extraction task. In particular, TagMe seems to obtain better performances on short texts (at the cost of stability) while Achemy seems more suitable for long documents such as scientific articles. We observed that KP-Miner and Maui have better stability than Alchemy when considering different domains, while TagMe seems very instable, especially on long documents.

There are few limitations we would like to highlight that might have an impact on the results presented in this study. One is related to the number of keywords returned by the semantic annotators. In particular, Zemanta only returns eight keywords for each document. This parameter cannot be changed and as a consequence, the performance of the system can be affected when evaluated on a gold standard that requires a bigger

[14] We contacted KP-Miner authors and they provided us with the latest version of their system. This version integrates a bug fix that was not taken into account during the official runs which explains a slightly better performance in this paper than in the competition.

number of extracted keywords. More generally, we noticed that some semantic annotators (namely TextRazor and OpenCalais) include generic keywords or categories that are not part of the original documents. This feature could be valuable for higher-level or more abstract information retrieval tasks but the downside is that these outputs are very difficult to evaluate as they are often not part of the annotated data provided by gold standards.

6 Conclusions and perspectives

Semantic annotators are commonly used to analyze documents and enrich their content with semantic annotations. Previous works have mainly focused on the evaluation of named entity recognition or named entity disambiguation. We presented an evaluation of the quality of semantic annotators in the context of the keyword extraction task. We considered five semantic annotators, Alchemy, OpenCalais, TagMe, TextRazor, Zementa as well as two keyword extractors KP-Miner and Maui.

The systems were evaluated using two keyword extraction datasets: SemEval and Crowd500. Our evaluation demonstrated that semantic annotators and keyword extractors achieve less than 30% F1-score on these datasets. On the SemEval dataset, the top three systems are KP-Miner, Alchemy Keyword and Maui (F1-score: 26.16% | 21.21% | 20.9%). On the Crowd500 dataset the top three systems are TagMe, Maui and KP-Miner (F1-score: 16.93% | 15.37%| 13.48%).

Following the global evaluation of the systems, we conducted a detailed assessment of the keywords extracted by the systems across different domains. We showed that systems tend to extract keywords successfully in domains with high keyword density values as the likelihood of finding a keyword is higher. Most of the current approaches rely on the frequency of the terms in documents. Consequently, keywords that are pertinent but rare in documents are more difficult to identify.

In our datasets, the high keyword density domains were World politics, US politics and Crime, while the low keyword density ones were Computer communication, Artificial intelligence and Information retrieval. The systems that perform best on these low keyword density domains were KP-Miner, Alchemy Keyword and Maui. Our domain analysis confirmed that TagMe was more suited to process small documents.

This study computed the F1-score based on an ”exact matching” constraint. In future work, we plan to refine our evaluation method by relaxing the matching constraints when comparing the extracted keywords to a gold standard. In our current process, a system is penalized if it does not identify the exact keyword contained in the gold standard. Overall, our main message is that semantic annotators need to leverage keyword extraction research to enhance their spotting phase. Semantic Web development necessitates quality annotations and the performance of semantic annotators will play an important role in enabling or slowing down a useful Web of Data.

7 Acknowledgements

The authors thank Pr. Samhaa R. El-Beltagy for providing the lastest version of the KP-Miner system as well as guidance for using the system. This research was partly funded by the NSERC discovery grants program.

Bibliography

[1] Matsuo, Y., Ishizuka, M.: Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools (2004)

[2] Stuart, R., Dave, E., Nick, C., Wendy, C.: Automatic Keyword Extraction from Individual Documents. In: Text Mining. John Wiley & Sons, Ltd (2010) 1–20

[3] El-Beltagy, S.R., Rafea, A.: Kp-miner: A keyphrase extraction system for english and arabic documents. Information Systems (2009)

[4] Rao, W., Chen, L., Hui, P., Tarkoma, S.: Move: A large scale keyword-based content filtering and dissemination system. In: IEEE 32nd ICDS. (2012)

[5] Yih, W.t., Goodman, J., Carvalho, V.R.: Finding advertising keywords on web pages. In: Proceedings of WWW '06. (2006)

[6] Yang, S., Jin, J., Parag, J., Liu, S.: Contextual advertising for web article printing. In: Proceedings of DocEng '10. (2010)

[7] Vidal, M., Menezes, G.V., Berlt, K., de Moura, E.S., Okada, K., Ziviani, N., Fernandes, D., Cristo, M.: Selecting keywords to represent web pages using wikipedia information. In: Proceedings of WebMedia '12. (2012)

[8] Bunescu, R., Pasca, M.: Using encyclopedic knowledge for named entity disambiguation. In: EACL-06. (2006)

[9] Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to wikipedia. In: Proceedings of the 49th Annual Meeting of the ACL-HLT. (2011)

[10] Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T.: Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In: Proceedings of the 5th International Workshop on Semantic Evaluation. (2010)

[11] Kim, S., Medelyan, O., Kan, M.Y., Baldwin, T.: Automatic keyphrase extraction from scientific articles. Language Resources and Evaluation (2013)

[12] Marujo, L., Gershman, A., Carbonell, J., Frederking, R., ao P. Neto, J.: Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. In: Proceedings of LREC. (2012)

[13] Medelyan, O., Frank, E., Witten, I.H.: Human-competitive tagging using automatic keyphrase extraction. In: Proceedings of EMNLP '09. (2009)

[14] Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of EMNLP '03. (2003)

[15] Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G.: Kea: practical automatic keyphrase extraction. In: Proceedings of DL '99. (1999)

[16] Turney, P.D.: Learning algorithms for keyphrase extraction. Inf. Retr. (2000)

[17] Zhang, C., Wang, H., Liu, Y., Wu, D., Liao, Y., Wang, B.: Automatic keyword extraction from documents using conditional random fields. Journal of Computational Information Systems (2008)
[18] Hulth, A.: Enhancing linguistically oriented automatic keyword extraction. In: Proceedings of HLT-NAACL 2004: Short Papers. (2004)
[19] Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Proceedings of EMNLP 2004. (2004)
[20] Grineva, M., Grinev, M., Lizorkin, D.: Extracting key terms from noisy and multitheme documents. In: Proceedings of WWW '09. (2009)
[21] Tsatsaronis, G., Varlamis, I., Nørvåg, K.: Semanticrank: Ranking keywords and sentences using semantic graphs. In: Proceedings of the 23rd International Conference on Computational Linguistics. (2010)
[22] Lahiri, S., Choudhury, S.R., Caragea, C.: Keyword and keyphrase extraction using centrality measures on collocation networks. CoRR (2014)
[23] Qureshi, M.A., O'Riordan, C., Pasi, G.: Short-text domain specific key terms/phrases extraction using an n-gram model with wikipedia. In: Proceedings of ACM CIKM. (2012)
[24] Joorabchi, A., Mahdi, A.E.: Automatic subject metadata generation for scientific documents using wikipedia and genetic algorithms. In: Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management. (2012)
[25] Hachey, B., Radford, W., Nothman, J., Honnibal, M., Curran, J.: Evaluating entity linking with wikipedia. In: Artificial Intelligence. (2013)
[26] Rizzo, G., Troncy, R., Hellmann, S., Bruemmer, M.: NERD meets NIF: Lifting NLP extraction results to the linked data cloud. In: Linked Data on the Web (LDOW2012). (2012)
[27] Cornolti, M., Ferragina, P., Ciaramita, M.: A framework for benchmarking entity-annotation systems. In: Proceedings of WWW '13. (2013)
[28] Rizzo, G., van Erp, M., Troncy, R.: Benchmarking the extraction and disambiguation of named entities on the semantic web. In: LREC, Reykjavik, ICELAND (2014)
[29] Lopuszynski, M., Bolikowski, L.: Tagging scientific publications using wikipedia and natural language processing tools. comparison on the arxiv dataset. CoRR (2013)
[30] Gagnon, M., Zouaq, A., Jean-Louis, L.: Can we use linked data semantic annotators for the extraction of domain-relevant expressions? In: Proceedings of WWW '13 Companion. (2013)
[31] Steinmetz, N., Knuth, M., Sack, H.: Statistical analyses of named entity disambiguation benchmarks. In: NLP-DBPEDIA@ISWC. (2013)
[32] Ferragina, P., Scaiella, U.: Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In: Proceedings of ACM CIKM. (2010)