An analysis of the semantic annotation task on the linked data cloud

Michel Gagnon*

Department of Computer Engineering and Software Engineering,
Polytechnique Montréal, C.P. 6079, succ. Centre-ville,
H3C 3A7, Montréal QC, Canada
Email: michel.gagnon@polymtl.ca
*Corresponding author

Amal Zouaq

Department of Computer Engineering and Software Engineering,
Polytechnique Montréal, C.P. 6079, succ. Centre-ville,
H3C 3A7, Montréal QC, Canada
and
School of Electrical Engineering and Computer Science,
University of Ottawa,
800 King Edward Ave.,
K1N 6N5, Ottawa, Ontario, Canada
Email: amal.zouaq@polymtl.ca

Francisco Aranha

Fundaçao Getulio Vargas,
Escola de Administraçao de Empresas de Sao Paulo,
Rua Itapeva, 474 - 9o andar, cep 01332-000 - Bela Vista,
Sao Paulo, SP, Brazil
Email: chico.aranha@gmail.com

Faezeh Ensan

Ferdowsi University of Mashhad,
Mashhad Iran,
and
University of New Brunswick,
Fredericton, Canada
Email: ensan@um.ac.ir

Ludovic Jean-Louis

Netmail, 180 Peel Street,
Suite 333, H3C 2G7,
Montreal, QC, Canada
Email: ludovic.jean-louis@netmail.com

Abstract: Semantic annotation, the process of identifying key phrases in texts and linking them to concepts in a knowledge base, is an important basis for semantic information retrieval and the semantic web uptake. Despite the emergence of semantic annotation systems, very few comparative studies have been published on their performance. In this paper, we provide an evaluation of the performance of existing systems over three tasks: full semantic annotation, named entity recognition, and keyword detection. More specifically, the spotting capability (recognition of relevant surface forms in text) is evaluated for all three tasks, whereas the disambiguation (correctly associating an entity from Wikipedia or DBpedia to the spotted surface forms) is evaluated only for the first two tasks. We use logistic regression to identify significant performance differences. Although some of the annotators are specifically targeted at some task (NE, SA, KW), our results show that they do not necessarily obtain the best performance on those tasks. In fact, systems identified as full semantic annotators beat all other systems on all data sets. We also show that there is still much room for improvement for the identification of the most relevant entities described in a text.

Keywords: semantic annotation; linked data cloud; performance evaluation.

Reference to this paper should be made as follows: Gagnon, M., Zouaq, A., Aranha, F., Ensan, F. and Jean-Louis, L. (2019) 'An analysis of the semantic annotation task on the linked data cloud', *Int. J. Metadata, Semantics and Ontologies*, Vol. 13, No. 4, pp.317–329.

Biographical notes: Michel Gagnon is Professor at the Computer Engineering Department of Polytechnique Montreal since 2002. Previously, he worked as a team leader at Machina Sapiens Inc. for the development of grammar checkers, and as a professor at the Univerdade Federal do Paraná, in Brazil. He received his PhD degree in computer science in 1993 from the Université de Montréal. Since then, he has been working on natural language processing, with a special attention to semantics. His research activities also include the semantic web. Currently, he is co-leader of WeST lab, whose main activities are related to the extraction of knowledge from texts.

Amal Zouaq is an Associate Professor at Ecole Polytechnique de Montreal and a member of IVADO. Her research interests include natural language processing, semantic web, ontology engineering, knowledge extraction and educational data mining. She serves as a member of the program committee of various conferences and journals on natural language processing, semantic web, knowledge engineering and educational data mining.

Francisco Aranha, PhD, MSc, MBA, is a full professor at the Department of Technology and Data Science (TDS) at the São Paulo School of Business (FGV-EAESP). Presently Associate Dean for the Center of Teaching and Learning Advancement, he has acted as Associate Dean for the Undergraduate Program in Business Administration and for the Master and Doctorate Programs in Business Administration.

After two postdoctoral fellows at the University of British Columbia and Arthabasca University, Faezeh Ensan is now working as Assistant Professor at Ryerson University and as Adjunct-Assistant Professor at Ferdowsi University of Mashhad.

Ludovic Jean-Louis is a research engineer at Netgovern Inc. His current research interests focus on leveraging machine learning and information retrieval techniques to process large heterogeneous data repositories. Before this position, he was a postdoctoral researcher at Polytechnique Montreal. He obtained a PhD in computer science in 2011, from the University of Paris XI, under the supervision of Dr. Olivier Ferret and Dr. Romaric Besancon. Particular research interests include knowledge base population, semantic similarity, micro-services, document classification, etc.

1 Introduction

Semantic annotation is an important basis for realising the semantic web vision (Dill et al., 2003; Shen et al., 2015), a vision of a web of machine-understandable data, and an important foundation for retrieving semantic information. Semantic annotation involves the recognition of short text fragments called *mentions* in documents (aka *spotting)* and links them to URIs defined in a knowledge base, aka *disambiguation.* Originally, automatic semantic annotation has been implemented using well-defined and restricted ontologies and knowledge bases (Kiryakov et al., 2011). This led to several platforms such as KIM (Kiryakov et al., 2011) or Apache Stanbol (Sinaci and Gonul, 2012). The emergence of the linked data cloud has encouraged the development of several annotation services (Milne and Witten, 2013; Ferragina and Scaiella, 2010; Mendes et al., 2011) such as DBpedia Spotlight and Yahoo which exploit LOD data sets and especially DBpedia/Wikipedia (Bizer et al., 2009) as their background knowledge bases. These knowledge bases, with their wide coverage, their structured description of content and their dynamic nature, are well suited for enriching almost all types of unstructured text. However, they also raise new challenges due to their size and their cross-domain nature. Thus, it is not surprising that, among the various services that appeared in the last few years, we see a great variation in terms of performance (Cornolti et al., 2013a; Jean-Louis et al., 2014; Chen et al., 2013; Ruiz and Poibeau, 2015; Derczynski et al., 2015; Gangemi, 2013). Additionally, mentions in text might represent entities, concepts, keywords, multi-word expressions, events, etc. and depending on the task at hand, some types of mentions might be more appropriate. While the majority of linked data annotators are described as "Semantic Annotators" without any specific type of mention in mind, in practice, many are more geared towards named entities (e.g. organisations, people) than topics or keywords (e.g. artificial intelligence) for instance. It is thus often difficult to distinguish the most adequate service among the plethora of available web APIs. In this paper, our aim is to facilitate such a choice, formalise semantic annotation tasks as well as assess some of the available linked data semantic annotators' strengths and weaknesses for these tasks. Note that this paper does not aim at providing an exhaustive survey of existing annotation APIs but rather focuses on some prominent APIs and describes a methodology for the evaluation of semantic annotators.

Based on our analysis of the state of the art, we identified that semantic annotators can be applied to three main tasks:

Traditional Semantic Annotation (SA): Given a particular knowledge base, SA consists of the identification of all the possible KB entities in a document. Here, mentions can represent keywords, classes, individuals and might be of any type. The early semantic annotation platform KIM (Kiryakov et al., 2011) is a good example of such an approach, which is often based on the assumption of a closed knowledge base. In linked data-based semantic annotators, mostly those based on DBpedia/Wikipedia, all Wikipedia content (aka resources) can be identified in documents.

Named entity annotation (NE): The second task focuses on the annotation of named entities, which refer to individuals of certain types. Named entity annotation is an extension of the simpler task of Named Entity Recognition (NER), an important topic in natural language processing that has been vastly studied and investigated in the literature (Nadeau and Sekine, 2007). The main difference is that traditional NER has very limited types such as PERSON and ORGANISATION which are generally not defined in an ontology. On top of these traditional named entities, current linked data-based annotators define an extended range of named entities and rely on a finer classification of each named entity (e.g. politicians, poets and non-governmental organisations).

Keyword extraction (KW): The third task can be described as the identification of a limited number of *prominent* domain-related key phrases and concepts. An example would be the extraction of key phrases related to a specific research topic in academic publications (Qureshi et al., 2012) or the identification of biologically significant phrases related to protein functions (Andrade and Valencia, 1998). This task requires filtering and ranking capabilities that identify the most important mentions. Compared to traditional keyword extractors, linked data semantic annotators can also (not always) link the extracted keywords to their corresponding concepts in a knowledge base.

These three tasks are used as a basis for evaluating and predicting the performance of some of the most prominent semantic annotators on similar data sets. While there are frameworks such as GERBIL (Usbeck et al., 2015) which handle the evaluation of semantic annotators, none has made the distinction based on the three tasks as described above. In recent extensions (Waitelonis et al., 2016, 2019), GERBIL provides ways to better evaluate systems specialised in some domains or some types of entities, but still does not provide ways of clearly distinguishing among the three tasks. Additionally, our results show that it does not suffice to compare metrics' results to evaluate the interest of semantic annotators. As we will see in this paper, a finer statistical analysis indicates that some semantic annotators' results are indistinguishable.

The remainder of this paper is organised as follows: in Section 2, we briefly present the limited state of the art on semantic annotators' evaluation. Section 3 describes our research methodology, including our research question, data sets and evaluation metrics. We also provide a description of the evaluated annotators. The following two sections describe our experimental results, first by estimating the performance of the systems on all data sets and then by considering only the three categories of systems (SA, NE, KW) instead of their individual performances. Section 6 discusses our findings and the limitations of this study, and concludes with further discussion on the evaluation strategies and results.

2 State of the art

Very few comparative studies have been published on the performance of semantic annotation systems (Cornolti et al., 2013b; Joksimovic et al., 2013), especially in the three tasks mentioned above. Existing evaluation results are mostly related to specific semantic annotation services (e.g. Mendes et al. (2011) and Ferragina and Scaiella (2010)), and hence are based on diverse metrics and gold standards, different data gathering methodologies and a limited set of evaluation data sets. In general, these works do not include numerous annotation systems for their evaluation and comparison. Two significant exceptions are the works reported by Cornolti et al. (2013b) and Usbeck et al. (2015). In these works, the authors provide a framework for benchmarking semantic annotation systems and comparing their performance. They introduce a set of tasks for which semantic annotation systems are usually employed (e.g. Annotate to Wikipedia (A2W) and Disambiguate to Wikipedia (D2W)) and provide metrics to evaluate systems in these contexts. However, these results might not be sufficient to distinguish the top performing annotators without a deeper statistical analysis. In this paper, the selected corpora are considered as samples of the population of similar documents used to estimate the performance of the semantic annotation systems. We only consider two annotators to perform differently if their respective confidence intervals for average performance do not overlap. In other words, we consider that two systems can show different point estimates for performance in the sample data (the corpora), and still have equal performance when all possible similar data are used (population). For them to be considered different, they must present significant differences in performance estimates.

One of the main limitations of existing literature in semantic annotation evaluation (Cornolti et al., 2013b; Meij, 2013) is that it does not take into account the fact that the performance of a system may vary according to a specific task. By contrast, our evaluation aims at providing experimental results on the performance of current semantic annotation systems for the three tasks (SA, NE, KW) with the objective of identifying annotators that are best suited for each of them. To achieve this, we rely on standard data sets that are experimentally selected for each task.

3 Research methodology

In this paper, we address the following research question:

> *RQ: How do linked data annotators perform on the three tasks (SA, NE and KW)?*

To answer this research question, we examine the overall spotting and disambiguation performance of linked data annotators in terms of precision and recall (these metrics are defined in Section 4).

3.1 Data sets

We selected three different groups of data sets in English (a *data set* is a corpus where mentions are spotted and disambiguated according to a gold standard). Each data set is focused on at least one of three tasks[1] and uses DBpedia/ Wikipedia as a background knowledge base. These data sets include:

1) AI and IITB for the semantic annotation task,
2) MSNBC, for the evaluation of named entity annotation, and
3) The SemEval and Inspec data sets, which are used for the evaluation of keyword extraction.

3.1.1 AI

The AI corpus is a small set of documents composed of Wikipedia articles related to the artificial intelligence domain, which was used in previous experiments for ontology learning from text (Jean-Louis et al., 2014). The gold standard was created by running all selected semantic annotators, evaluating the returned annotations as correct or incorrect, and then filtering out all the incorrect ones. This evaluation was performed by two of the authors of this paper (two postdoctoral researchers at the time of the experiment). Here, we assume that the union of the correct annotations provided by several different systems results in a coverage that is close to exhaustivity.

3.1.2 IITB

IITB is a data set proposed by Kulkarni et al. (2009) which includes more than a hundred documents comprehensively annotated by human experts. Documents were collected from popular websites on sport, entertainment, health and science. In the literature, IITB is often used for the evaluation of named entity annotation, but in our evaluation we associate it to the semantic annotation task, since it contains annotations that go beyond named entities (e.g.: *sniper, militant, October 7, president of Afghanistan).*

3.1.3 MSNBC

MSNBC is a small collection of news documents (18 documents) on different popular subjects such as sport, politics and technologies and was proposed by Cucerzan (2007). MSNBC is mainly focused on important named entities. However, an initial analysis revealed significant problems in the data set, such as entities that are indicated in the gold standard, but not found in the documents, entities cited in the documents, but absent from the gold standard, and, less frequently, incorrect entities specified in the gold standard. For the purpose of this research, we completely re-annotated the documents of this corpus, to obtain a gold standard more accurate than the original one.

3.1.4 SemEval

The SemEval data set (Kim et al., 2010) is a standard benchmark for keyword extraction that associates key phrases to documents. It contains 244 scientific articles, usually composed of six to eight pages. The articles cover different research areas of the ACM classification: Distributed Systems, Information Search and Retrieval, Distributed Artificial Intelligence, Multi-agent Systems, Social and Behavioural Sciences and Economics. Most articles essentially cover the Computer Science domain (75% of the documents) and the other documents cover the Economy domain. The gold standard includes key phrases assigned by annotators (75%) as well as key phrases assigned by the papers' authors (25%). The SemEval corpus is divided into a training data set (144 articles, 2070 key phrases) and testing data set (100 articles, 1443 key phrases). In our experiment, we consider the 244 articles as a single corpus.

3.1.5 Inspec

Inspec is a set of 2000 documents and consists of abstracts from scientific journal papers. Each abstract has two sets of keywords assigned by a professional indexer. One is a set of controlled terms from the Inspec thesaurus, and the other one is an uncontrolled set of keywords that contains any suitable term identified by the indexer in texts. Both sets can contain keywords that are not found in the abstracts. In our evaluation, we used only the uncontrolled set of keywords.

Table 1 shows some descriptive statistics about the number of documents and the number of mentions in each data set. We can notice that the Inspec data set contains the highest number of mentions (it also contains many more documents), but the fewest number of mentions per document (due to the smaller size of documents and due to the fact that the keyword extraction task identifies the most relevant keywords only). In the IITB and AI data sets, the average number of mentions per document is much higher than in the three other data sets. This is expected for the task of semantic annotation. Finally, the average number of words in SemEval is much higher than in other data sets, but this value is somehow misleading, since these documents contain a high number of tokens that are not words (for example, elements of mathematical formulas).

Table 1 Statistics on data sets

Corpus	*# doc*	*# words/ doc*	*# mentions*	*# mentions/ doc*	*Task*
AI	8	1322	713	89.1	SA
IITB	104	640	6866	66.0	SA
MSNBC	18	544	392	21.8	NE
SemEval	244	8022	3689	15.1	KW
Inspec	2000	124	19,244	9.6	KW

3.2 Semantic annotators

In this section, we briefly present the semantic annotators selected for this study. For the purpose of our evaluation, we selected academically or industrially prominent semantic annotators available through a web API.

Table 2 shows all the evaluated annotators categorised according to their best-suited task based on their description: semantic annotation (SA), named entity annotation (NE) or keyword annotation (KW). We also indicate if the service is commercial, if the result of the annotation process may contain external entities that are not found in the text and, finally, the knowledge base that is used to disambiguate the entities.

Table 2 Systems used in the current study

System	*Cat.*	*Commerc.*	*External ent.*	*KB*
Watson/SA	SA	✓	✓	DBpedia
Aylien/SA	SA	✓		DBpedia
Babelfy/SA	SA			DBpedia, Babelnet
Dandelion	SA	✓		Wikipedia,
Spotlight	SA			DBpedia
Open Calais	SA	✓		Proprietary
Tagme	SA			Wikipedia
Umbel	SA			Umbel
Yahoo	SA	✓		Wikipedia
Ambiverse	SA	✓		Wikipedia
Aylien/NE	NE	✓		–
Babelfy/NE	NE			DBpedia, Babelnet
Enrycher/NE	NE			DBpedia, YAGO, OpenCyc
MeaningCloud/ NE	NE	✓		DBpedia
TextRazor	NE	✓		Wikipdia, Freebase,
Watson/NE	NE	✓		DBpedia
Aylien/KW	KW	✓		–
Enrycher/KW	KW		✓	–
MeaningCloud/ KW	KW	✓		–
Watson/KW	KW	✓		–

Hereafter, we describe the chosen semantic annotators. In some cases, the description is very brief due to the lack of published research on the semantic annotator.

Watson[2] APIs employ a set of deep linguistic parsing methods and statistical language processing techniques for performing semantic annotation. Various APIs are available, among which three are relevant to our research objectives: named entity extraction (Watson/NE), keyword extraction (Watson/KW), and concept extraction (Watson/SA). The named entity extractor (Watson/NE) is able to disambiguate the detected entities and resolve co-references. Entities are linked to various data sets on the Linked Open Data Cloud (LOD). Keyword extraction (Watson/KW) produces a list of key phrases without any linkage to an external knowledge base (i.e. without disambiguation). Concept extraction (Watson/SA) produces a list of concepts, that is, topics that are not necessarily mentioned in the text, along with their corresponding links on the LOD.

Aylien[3] is another commercial product that offers two services that are relevant for our study. One is the concept extraction service (Aylien/SA) and the other is the entity extraction service, which not only extracts named entities, but also keywords. Since these results correspond to different tasks in our framework, we analysed them separately (Aylien/NE and Aylien/KW). Note that the second service does not provide any disambiguation for the annotated entities.

On its website[4], Babelfy (Moro et al., 2014) is defined as "a unified, multilingual, graph-based approach to Entity Linking and Word Sense Disambiguation based on a loose identification of candidate meanings coupled with a densest sub-graph heuristic which selects high-coherence semantic interpretations". Babelfy is based on BabelNet, a multilingual semantic network. In our evaluation, we used two services: one for semantic annotation (Babelfy/SA) and one for named entities (Babelfy/NE).

Dandelion[5] offers several text analysis services for many languages: entity extraction, text similarity, text classification, language detection and sentiment analysis. Only the first one is of interest for our study.

Open Calais[6] is a service offered by Thomson Reuters. It can detect different kinds of entities, which are disambiguated with a proprietary knowledge base. It can also detect events, relationships and topics.

DBpedia Spotlight[7] (Mendes et al., 2011) is a configurable annotator that is linked to DBpedia (we used the default settings). After the spotting phase, DBpedia Spotlight pre-ranks DBpedia concept candidates for each spotted key phrase in text. It uses a similarity score to determine which candidate concept is the most relevant. The similarity score takes into account the context of the phrase (a window of words around the phrase) and the context of each candidate concept.

Tagme[8] is a semantic annotator mainly designed for analysing short texts such as tweets (Ferragina and Scaiella, 2010), but it has also been reported to perform well on longer documents (Cornolti et al., 2013a). Tagme tokenises a given text and finds candidate spots from token sequences of up to six words. It uses a set of heuristics and probability and coherence measures to decide which spotted candidates should be considered for disambiguation and which spot must be pruned from the result set. Tagme returns all annotations in a text plus their corresponding relevance scores according to the text topic.

Umbel[9] offers two tagging services. One tries to detect concepts from the Umbel ontology in texts, and the other is restricted to noun phrases. It is the latter that has been used in this study. Note that by default, Umbel does not apply any stemming.

Yahoo Content Analysis API[10] annotates entities and concepts and also provides a ranking of these entities and concepts, according to their overall relevance. Access to the service is achieved through the Yahoo Query Language (YQL), an SQL-like language that enables querying, filtering, and combining data across the web.

TextRazor[11] offers many services for the extraction of information from text. It also enables customisation by using Prolog rules. In our study, we only use the entity recognition service.

Enrycher[12] provides deep and shallow text processing services. We used the two following services: named entity resolution (Enrycher/NE) and keyword detection (Enrycher/ KW).

Finally, MeaningCloud[13] offers several text analysis services for many languages: topic extraction, text classification, sentiment analysis and text clustering. The topic extraction service detects and disambiguates named entities, and it can also extract keywords. MeaningCloud offers the possibility of adding your own dictionaries in its annotation services.

4 Evaluation of annotators' performances

The metrics that are of interest for the evaluation of the semantic annotators are the following ones:

- *Precision:* the ratio of the number of correct items returned by the annotator over the total number of items returned by the annotator.
- *Recall:* the ratio of the number of correct items returned by the annotator over the total number of items specified in the gold standard.

In our evaluation of systems, as explained in the following sections, we estimate the values of these metrics for the overall population of similar documents. We ran all semantic annotators on all the available data sets. Table 3 shows the total number of annotations extracted by each semantic annotator. We can notice considerable variations across systems and across data sets. For example, Tagme returns 5.7 times more annotations than DBpedia Spotlight, on average (41834 versus 7276). Similarly, Babelfy/SA, Dandelion, MeaningCloud/KW, Tagme, TextRazor and Watson/KW return more than 10,000 annotations on the average, while other systems return a much lower number of annotations (1289 for Enrycher/NE, and 2250 for Babelfy/NE). We may expect here that systems with the largest number of annotations will exhibit a high recall. We can also note that many more annotations are extracted from SemEval compared to other data sets. In fact, in this corpus, there are much more extracted keywords on the average than the number of correct mentions in the gold standard (more than 20,000 on average, compared to 3689 in the gold standard). We thus expect a very low precision for this corpus. AI and MSNBC are the smallest data sets in terms of numbers of annotations, which is expected, since the number of correct mentions in their gold standards is also very low compared to other data sets.

We evaluate the performance of the systems for the two main steps of semantic annotation, namely the spotting step and the disambiguation step. In each step, we analyse the precision and recall of semantic annotators for the SA and NE tasks. In the KW task, there is not any disambiguation step, so only spotting is evaluated. All mentions returned by annotators and all the ones provided in gold standards are stemmed using an implementation of the Porter stemmer. As an example, two key phrases 'parallel processes' and 'parallel processing' are matched to the gold standard entry "parallel process" because both have the same stem. This approach seems reasonable for the spotting phase, as we consider all these alternatives as valid mentions. If an entity is spotted more than once in a text, it appears only once in the gold standard. This means that we evaluate the capability of spotting at least one occurrence of each relevant mention.

Table 3 Number of annotations extracted by each system, for each corpus

System	*AI*	*IITB*	*MSNBC*	*SemEval*	*Inspec*	*Average*
Ambiverse	76	1716	173	12,433	2791	3438
Aylien/SA	260	2022	152	11,860	5241	3907
Aylien/NE	68	1332	147	12,283	2726	3311
Aylien/KW	143	2049	198	4607	37,886	8977
Babelfy/SA	1346	11,853	932	44,597	58,843	23,514
Babelfy/NE	113	2395	213	5132	3398	2250
Dandelion	1135	5464	402	62,580	33,359	20,588
Enrycher/KW	51	663	71	1119	14,445	3270
Enrycher/NE	54	1031	111	2996	2253	1289
MeaningCl oud/NE	115	1988	169	16,547	3820	4528
MeaningCloud/ KW	656	4983	374	38,754	25,605	14,074
Open Calais	159	2286	222	16,952	5345	4993
Spotlight	377	2933	210	22,821	10,040	7276
Tagme	1974	15,212	1079	120,291	70,616	41,834
TextRazor	989	6284	624	67,826	6498	16,444
Umbel	444	3184	257	26,529	17,803	9643
Watson/SA	63	809	79	1835	12,917	3141
Watson/KW	333	4710	393	10,592	40,346	11,275
Watson/NE	84	2191	203	9507	3198	3037
Yahoo	68	932	95	2040	12,979	3223
Average	425	3702	305	24,565	18,505	

4.1 The spotting step

In this evaluation, we are interested in estimating the probability that a system behaves correctly for a specific mention in a document. More precisely, to estimate the precision, we take each mention spotted by a system, and we calculate the probability of this mention to be correct. Similarly, for recall, we take each mention from the set of mentions in the gold standard, and we consider the probability for the system to detect this mention. In our data, since each correctly detected mention is associated to a value 1, and 0 otherwise, we can estimate these probabilities using logistic regression[14].

Tables 4 and 5 present the precision and recall estimates based on all data sets, considering all spotted entities returned by systems (without any filtering). Each column was calculated separately. For example, for the AI column, we used the set of all mentions spotted by all annotators on the AI corpus. Each observation in the database represents one mention and is characterised by two variables: *system* and *correctness*. *System* is the explanatory variable and *correctness* is the response variable in a logistic model; precision and recall values presented in the table are the parameter estimates, in decreasing order. The other columns were calculated similarly, one corpus at a time.

Table 4 Estimated precision of spotting, for each system and corpus. Horizontal lines separate indistinguishable groups of systems

AI		*IITB*		*MSNBC*		*SemEval*		*Inspec*	
Watson/SA	0.92	Aylien/SA	0.75	Ambiverse	0.73	Yahoo	0.26	Yahoo	0.33
Yahoo	0.79	Spotlight	0.69	Enrycher/NE	0.7	Watson/SA	0.14	Aylien/SA	0.31
Aylien/SA	0.69	Dandelion	0.63	Aylien/NE	0.65	Watson/KW	0.1	OpCalais	0.3
Spotlight	0.63	Enrycher/NE	0.60	Aylien/SA	0.57	Aylien/SA	0.044	Spotlight	0.24
Watson/NE	0.57	Yahoo	0.52	MCloud/NE	0.52	Spotlight	0.03	Watson/KW	0.22
OpenCalais	0.53	Ambiverse	0.53	Watson/NE	0.51	Babelfy/SA	0.026	MCloud/NE	0.15
Ambiverse	0.53	TextRazor	0.50	Babelfy/NE	0.49	OpCalais	0.024	Dandelion	0.13
Watson/KW	0.51	Aylien/NE	0.48	Spotlight	0.46	Dandelion	0.02	Babelfy/NE	0.13
Babelfy/NE	0.51	OpCalais	0.47	OpCalais	0.42	Aylien/KW	0.019	Watson/NE	0.12
MCloud/NE	0.48	Watson/SA	0.45	Yahoo	0.36	Watson/NE	0.017	TextRazor	0.12
TextRazor	0.45	Watson/NE	0.45	Dandelion	0.3	MCloud/NE	0.016	Ambiverse	0.12
Aylien/KW	0.43	Babelfy/NE	0.44	Watson/SA	0.3	TextRazor	0.014	Aylien/NE	0.11
Dandelion	0.41	MCloud/NE	0.44	TextRazor	0.24	Aylien/NE	0.013	Babelfy/SA	0.083
Enrycher/NE	0.41	Aylien/KW	0.42	Watson/KW	0.14	Babelfy/NE	0.0099	Tagme	0.076
Aylien/NE	0.4	MCloud/KW	0.38	Aylien/KW	0.13	Enrycher/KW	0.0085	Enrycher/NE	0.071
Babelfy/SA	0.29	Babelfy/SA	0.37	Tagme	0.085	Tagme	0.0085	Watson/SA	0.067
Enrycher/KW	0.24	Umbel	0.36	Babelfy/SA	0.054	Umbel	0.0083	Aylien/KW	0.057
Tagme	0.23	Tagme	0.32	MCloud/KW	0.012	MCloud/KW	0.0082	MCloud/KW	0.043
Umbel	0.2	Watson/KW	0.21	Enrycher/KW	0.0077	Ambiverse	0.0079	Umbel	0.021
MCloud/KW	0.19	Enrycher/KW	0.15	Umbel	0.0039	Enrycher/NE	0.0071	Enrycher/KW	0.0096

Table 5 Estimated recall of spotting, for each system and corpus. Horizontal lines separate indistinguishable groups of system

AI		*IITB*		*MSNBC*		*SemEval*		*Inspec*	
Dandelion	0.65	Tagme	0.72	TextRazor	0.83	Dandelion	0.34	Watson/KW	0.46
Tagme	0.63	Babelfy/SA	0.64	Ambiverse	0.76	Babelfy/SA	0.33	Tagme	0.28
TextRazor	0.63	Dandelion	0.49	Dandelion	0.69	Watson/KW	0.31	Babelfy/SA	0.25
Babelfy/SA	0.55	TextRazor	0.46	Babelfy/NE	0.65	Tagme	0.28	Dandelion	0.22
Spotlight	0.34	Spotlight	0.3	Watson/NE	0.62	TextRazor	0.27	Yahoo	0.22
Aylien/SA	0.26	MCloud/KW	0.28	Spotlight	0.61	Spotlight	0.19	Spotlight	0.12
Watson/KW	0.25	Aylien/SA	0.22	Tagme	0.61	Yahoo	0.15	Aylien/KW	0.11
MCloud/KW	0.18	Umbel	0.17	Aylien/SA	0.55	Aylien/SA	0.15	Aylien/SA	0.084
Umbel	0.12	OpCalais	0.16	Aylien/NE	0.54	OpCalais	0.11	OpC alais	0.083
OpC alais	0.12	Babelfy/NE	0.15	OpCalais	0.54	MCloud/KW	0.087	MCloud/KW	0.057
Aylien/KW	0.087	Watson/KW	0.15	MCloud/NE	0.53	Watson/SA	0.073	Watson/SA	0.044
Watson/SA	0.084	Watson/NE	0.14	Enrycher/NE	0.46	MCloud/NE	0.067	TextRazor	0.039
Babelfy/NE	0.083	Ambiverse	0.13	Watson/KW	0.33	Umbel	0.06	MCloud/NE	0.029
MCloud/NE	0.079	MCloud/NE	0.13	Babelfy/SA	0.31	Watson/NE	0.044	Babelfy/NE	0.022
Yahoo	0.079	Aylien/KW	0.12	Yahoo	0.18	Aylien/NE	0.043	Watson/NE	0.02
Watson/NE	0.07	Aylien/NE	0.092	Aylien/KW	0.15	Ambiverse	0.025	Umbel	0.019
Ambiverse	0.056	Enrycher/NE	0.09	Watson/SA	0.13	Aylien/KW	0.024	Ambiverse	0.017
Aylien/NE	0.039	Yahoo	0.071	MCloud/KW	0.031	Babelfy/NE	0.013	Aylien/NE	0.016
Enrycher/NE	0.032	Watson/SA	0.054	Umbel	0.0061	Enrycher/NE	0.0077	Enrycher/NE	0.0082
Enrycher/KW	0.017	Enrycher/KW	0.015	Enrycher/KW	0.003	Enrycher/KW	0.0033	Enrycher/KW	0.007

Additionally, in each column, systems were grouped by precision, according to the Furthest Neighbour clustering algorithm (Jonhson and Wichern, 1992) defined as follows:

1 Create the first group and consider it the current group.

2 Assign the first system in the list to the first group. Consider the first system's confidence interval as the seed.

3 Take the next system and consider it the current system.

4 Check the intersection of the confidence interval of the current system and the seed.

5 If the intersection is not empty assign the current system to the current group; if it is empty, start the next group and make it the current group, assign the current

system to it, and assign the confidence interval of the current system to the seed.

6 Go back to step (3) and continue with the algorithm until you have processed the last system in the column.

In essence, this procedure creates groups of systems whose performance cannot be internally considered significantly different among themselves, and which must be considered different from at least one of the systems in the other groups.

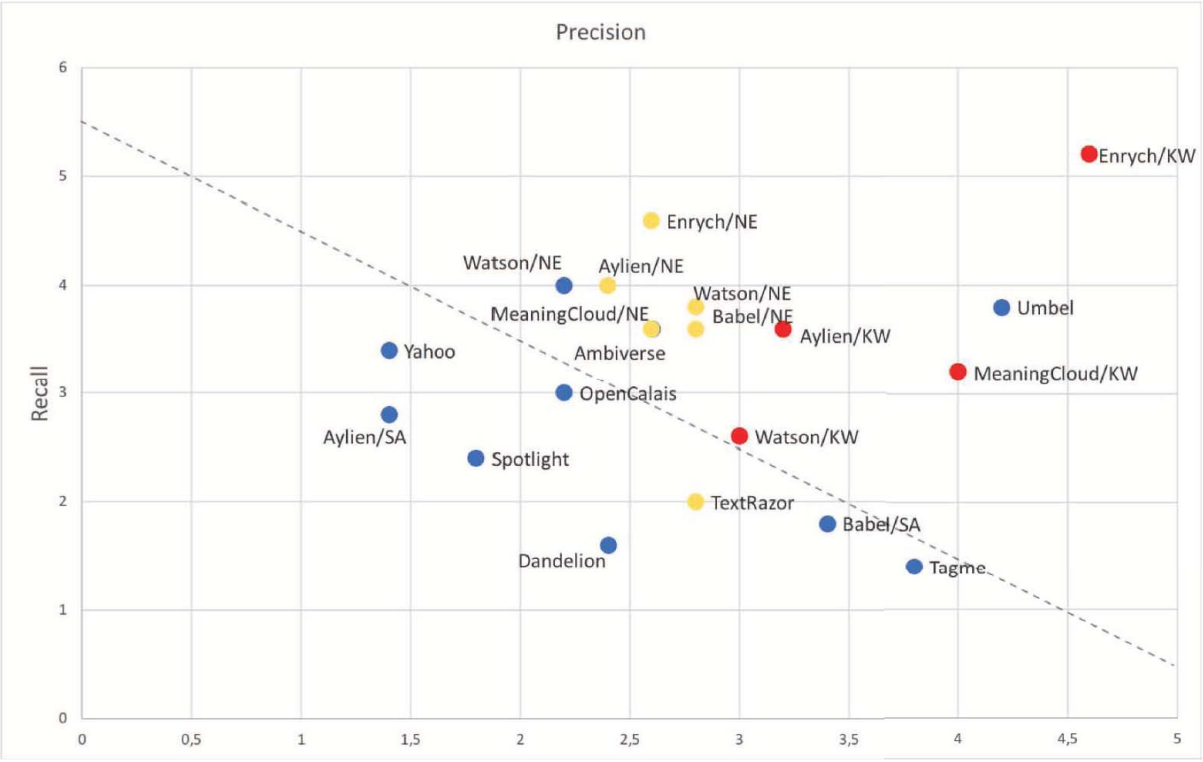
In Table 4, we can see that on the AI corpus, two systems display significantly better precision than the others: Watson/SA and Yahoo. On the IITB corpus, we observe a bigger number of groups of systems, with Aylien/SA and DBpedia Spotlight as the top-performing ones. Interestingly, the best systems are not the same ones as for the AI corpus, which corresponds to the same task (SA). We can also observe that the performance on IITB is lower than on AI. On the MSNBC corpus, which corresponds to the NE task, Ambiverse shares the leading position with three other systems. Among the four systems, only Enrycher/NE and Aylien/NE have been categorized as NE systems, the two other ones being SA systems. On SemEval and Inspec, the results are quite low, with Yahoo being among the top performers in both cases. Two systems are among the best ones (first or second group) on almost all corpora: Yahoo, Aylien/SA and DBpedia Spotlight. We can also note that, as expected, the precision observed on SemEval and Inspec is very low for all systems (remember that the SemEval corpus is the one with the largest number of annotations returned by the systems, thus increasing the probability of an incorrect annotation).

Table 5 presents the recall values. We can see that the best performing systems are not the same as in Table 4. For instance, Tagme and Babelfy/SA are among the worst systems in terms of precision, but among the best ones if we consider recall. This is expected, considering the high number of annotations returned by these systems, which necessarily favours recall over precision. Yahoo, which is among the best systems on both SemEval and Inspec in terms of precision, does not perform so well in terms of recall on these two same corpora. At the opposite, Watson/KW obtains low-precision values on these corpora, but performs very well in terms of recall. Note that on Inspec corpus, a recall of 0.46 puts this system clearly in a dominant position compared to the other ones.

4.2 *Semantic annotators ranking*

Based on our previous results, we propose a global ranking score that combines the individual rankings computed separately for each data set. These rankings are necessary to take into account groups rather than individual systems values, which are non-statistically distinguishable. Let C be one of our five data sets. A system s is attributed a local rank $i = Rank\ (s, C)$ if s is the i-th best group according to its estimated performance on data set C. The global ranking score is the average of system local rank i over the five data sets. Figure 1 shows the relative position of annotators by combining their ranks for precision (x axis) and recall (y axis). Note that the best ranking position is 1.

Figure 1 Rankings based on precision and recall for spotting task, across tasks and data sets (blue = SA systems, yellow = NE systems, red = KW systems)



If we consider precision only, Yahoo and Aylien/SA are the best annotators. In terms of recall, the best one is Tagme, closely followed by Dandelion. As expected, we can notice that annotators with good precision tend to have poor recall, and vice versa.

It is interesting to note that almost all NE and KW systems appear in the upper triangle, which means that their combined rankings in terms of precision and recall are not very good. At the opposite, most of the best systems, the ones situated in the left triangle, are SA systems.

4.3 The disambiguation step

In this section, we evaluate the disambiguation capability of semantic annotators. Three data sets are used for this task: AI, IITB and MSNBC. We also consider only full semantic annotators and named entity annotators, as keyword extractors usually do not return any disambiguation information with their key phrases. Also we restrict the evaluation to the systems that output links to Wikipedia or DBpedia. An annotation is considered correct if the pair (*m*, *e*), where *m* and *e* are the textual mention and the linked entity, respectively, is found as such in the gold standard. If the mention *m* is found in the gold standard but annotated with the wrong entity by the system, it is considered as a miss. If a mention returned by a system is not contained in the gold standard, it is ignored, since there is no way of determining whether the entity is correctly disambiguated. To check the validity of the disambiguated entity, we take into account Wikipedia redirect links. Thus, a system that would return *London_Heathrow_Airport* will be considered correct even if the entity in the gold standard is *Heathrow_Airport*. Put simply, we are computing the ratio of correctly disambiguated entities among the correctly spotted mentions, that is, we evaluate the probability P (Disambiguating | Spotting). Practically, this means that, for each annotator, mentions that are not correctly spotted by the annotator are removed from the gold standard, leading to a reduced gold standard. Here, recall is equal to precision, since all tested systems provide at least one entity for each spotted mention and these mentions are the same in the reduced gold standard.

4.4 Logistic regression for the disambiguation task

In this section, we repeat the same generalisation process using logistic regression for the disambiguation task (see Table 6). Remember that what is computed here is the probability *P*(Disambiguating | Spotting), that is, we estimate the probability of correctly disambiguating the entity when a mention is correctly spotted.

We can see that on AI, the performance of the systems is not distinguishable. On IITB, Watson/SA significantly dominates the other annotators, and it shares this position with five other systems on MSNBC. We also note that Tagme, which does not perform very well on IITB, is among the best ones on MSNBC.

4.5 Logistic regression for the full annotation process

Let us now consider the full annotation process, that is, the combination of spotting and disambiguation. In this case, an annotation is considered as a hit if it has been both correctly spotted and correctly disambiguated, and a miss if it has not been correctly spotted (and should be according to gold standard) or if it has been correctly spotted but incorrectly disambiguated. We see that the results are not as good (see Table 7). For example, precision values are much lower, especially on the IITB and MSNBC corpora.

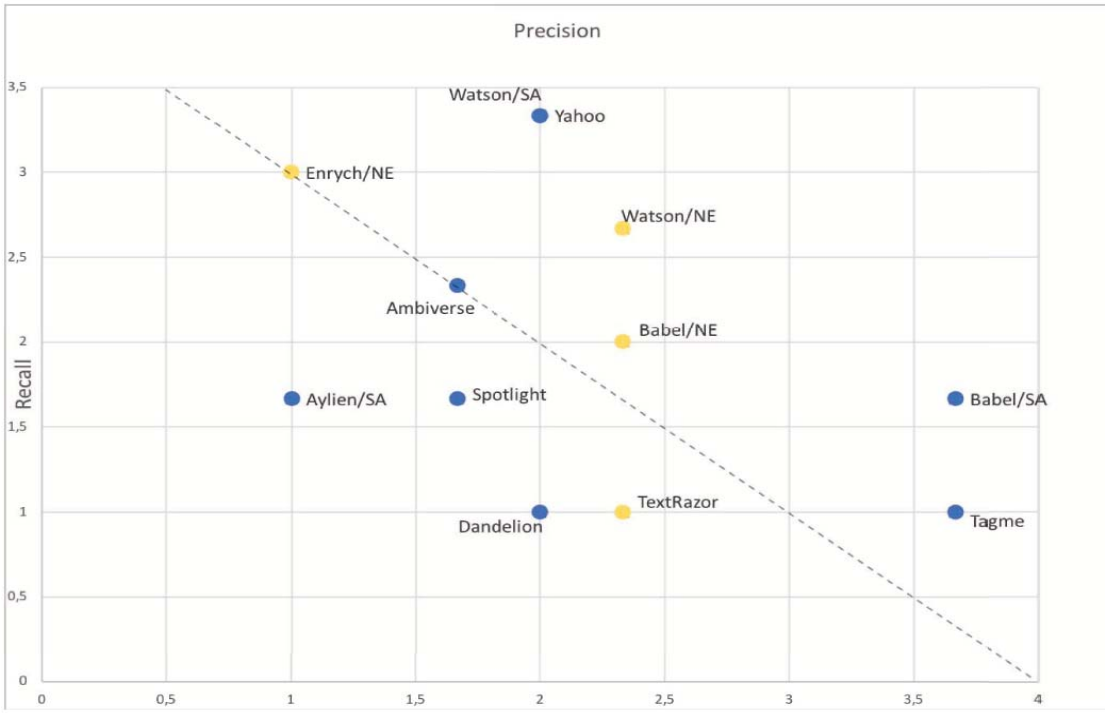
Table 6 Estimation of precision obtained for disambiguation task (superscript letters indicate the groups of systems that are not statistically distinguishable)

	AI	*IITB*	*MSNBC*
Watson/SA	1.00^{a}	0.95^{a}	0.95^{a}
Enrycher/NE	1.00^{a}	0.78^{b}	0.69^{b}
Ambiverse	0.91^{a}	0.82^{b}	0.80^{b}
TextRazor	0.91^{a}	0.76^{b}	0.81^{b}
Babelfy/NE	0.91^{a}	0.78^{b}	0.83^{a}
Tagme	0.91^{a}	0.61^{c}	0.87^{a}
Yahoo	0.89^{a}	0.74^{b}	0.76^{b}
Aylien/SA	0.88^{a}	0.76^{b}	0.83^{a}
Spotlight	0.86^{a}	0.75^{b}	0.82^{a}
Dandelion	0.86^{a}	0.71^{c}	0.87^{a}
Babelfy/SA	0.74^{a}	0.55^{d}	0.20^{c}
Watson/NE	0.62^{a}	0.59^{c}	0.60^{b}

Table 7 Estimation of precision and recall obtained for the full annotation process

Precision			
	AI	*IITB*	*MSNBC*
Watson/SA	0.84^{a}	0.23^{c}	0.26^{6}
Enrycher/NE	0.65^{a}	0.46^{a}	0.50^{a}
Aylien/SA	0.55^{a}	0.56^{a}	0.47^{a}
Ambiverse	0.52^{b}	0.43^{b}	0.58^{a}
Spotlight	0.50^{b}	0.51^{a}	0.37^{b}
Babelfy/NE	0.45^{b}	0.33^{c}	0.39^{b}
Yahoo	0.37^{b}	0.39^{b}	0.26^{b}
TextRazor	0.35^{b}	0.37^{b}	0.18^{c}
Watson/NE	0.31^{b}	0.26^{c}	0.31^{b}
Dandelion	0.30^{b}	0.43^{b}	0.24^{b}
Babelfy/SA	0.17^{c}	0.20^{d}	0.01^{d}
Tagme	0.19^{c}	0.19^{d}	0.067^{d}
Recall			
	AI	*IITB*	*MSNBC*
Dandelion	0.61^{a}	0.35^{a}	0.57^{a}
Tagme	0.60^{a}	0.43^{a}	0.50^{a}
TextRazor	0.59^{a}	0.38^{a}	0.64^{a}
Babelfy/SA	0.38^{a}	0.35^{a}	0.063^{c}
Spotlight	0.29^{b}	0.21^{b}	0.49^{a}
Aylien/SA	0.22^{b}	0.16^{b}	0.46^{a}
Watson/SA	0.079^{b}	0.029^{e}	0.13^{c}
BabelfyNE	0.0073^{b}	0.11^{c}	0.52^{a}
Ambiverse	0.061^{c}	0.10^{c}	0.60^{a}
Enrycher/NE	0.055^{c}	0.075^{d}	0.33^{b}
Yahoo	0.044^{c}	0.051^{d}	0.14^{c}
Watson/NE	0.040^{c}	0.079^{c}	0.37^{b}

Figure 2 Ranking based on precision and recall, for the full annotation process across tasks and data sets (blue = SA systems, yellow = NE systems)



Watson/SA, which performs well on all corpora for disambiguation, outperforms the other systems only on AI for the full annotation process. Aylien/SA and Enrycher/NE are among the top-performing systems in terms of precision for all corpora. In terms of recall, Dandelion, Tagme and TextRazor are among the top-performing systems for all three corpora. Another clear result is that some systems display much better recall on MSNBC than on the two other corpora: Watson/SA, Babelfy/NE, Ambiverse, Enrycher/NE and Watson/NE. In the case of Watson/SA and Ambiverse, this is somehow surprising, since they are classified as SA systems.

Figure 2 shows the combined rankings in terms of precision and recall using the same formula as before. Overall, we can draw the same conclusion about the superiority of SA systems.

To summarise, our results show that despite good disambiguation capabilities, semantic annotators poorly identify entity mentions in text (spots), which leads to a weak performance for the full annotation process.

5 Evaluation for the three tasks

In this section, we determine if there is some correlation between the task associated to an annotator (SA, NE or KW), and the task associated to each data set (remember that each data set is also associated to one of these three tasks). We computed a logistic regression where we consider all the systems associated with each of the three tasks. In our analysis we will consider both spotting, which concerns all three types of systems (SA, NE and KW), and full task (spotting + disambiguation), which concerns only SA and NE systems. Results are shown in Tables 8 and 9. The first observation is that SA annotators globally perform better at spotting than other systems on all data sets, if we consider both precision and recall. If we consider only the precision, KW systems do not perform as well as SA and NE systems, not even on the corpora where they should perform better (SemEval and Inspec). Another conclusion is that there is not any clear correspondence between the original task of a semantic annotator and the corresponding data set. For example, we cannot conclude that SA annotators perform better than others on the AI and IITB data sets, contrarily to our expectation. Similarly, NE annotators are not necessarily the best performing annotators on MSBNC (they display worse recall values) and, finally, KW systems do not dominate on the last two data sets.

Table 8 Estimated probabilities of correctly spotting an entity for each task and each data set

	AI	*IITB*	*MSNBC*	*SemEval*	*Inspec*
			Precision		
SA	0.3639	0.4267	0.2294	0.0189	0.1149
NE	0.3993	0.4637	0.2570	0.0219	0.1311
KW	0.3262	0.3864	0.2013	0.0161	0.0990
			Recall		
SA	0.2766	0.2891	0.5202	0.1657	0.1371
NE	0.0805	0.0852	0.1990	0.0435	0.0351
KW	0.2287	0.2397	0.4567	0.1335	0.1097

Table 9 Estimated probabilities of correctly spotting and disambiguating an entity for each task and each data set

	AI	*IITB*	*MSNBC*
	Precision		
SA	0.2795	0.2891	0.1698
NE	0.3540	0.3648	0.2241
	Recall		
SA	0.2983	0.2274	0.4582
NE	0.1219	0.0877	0.2164

Now considering the full task of spotting and disambiguating, Table 9 shows that once again, there is no clear match between the systems and the tasks. Contrarily to expectation, precision of NE systems is lower on MSNBC than on other corpora, and recall of SA systems is higher on MSNBC than on the two corpora that correspond to their task (AI and IITB).

6 Discussion and conclusions

The work presented in this paper provides a comprehensive study over a wide range of semantic annotation systems for three different tasks. This kind of evaluation provides a basis for choosing a particular system depending on the task at hand, e.g. achieving a comprehensive annotation of documents, identifying named entities, or providing a small number of relevant keywords. We provide a statistical analysis of our results through a logistic regression model. Hereafter, we refer to our initial research question and provide some answers drawn from our experiments.

How do linked data annotators perform on the three tasks (SA, NE and KW)?

The first major observation is the weak spotting capability of annotators across all three tasks: very few systems display good results for both precision and recall. The notable exception is Ambiverse on MSNBC (precision of 0.73 and recall of 0.76). The worse performances are observed on the KW task, where the highest values of precision and recall are 0.33 and 0.46, respectively. Once the spots are identified, the majority of annotators disambiguate the spotted mentions correctly (the best systems have precision close to 1.0). However, due to the low spotting performance, the performances for the full annotation process are not very good for most systems, especially in recall.

The second major observation is that semantic annotators perform better than the other two types of systems on all three tasks. Even when the identified task is a keyword extraction or a named entity recognition, SA systems are still the best choice, especially Alien/SA, Yahoo, DBpedia Spotlight and Dandelion. When we statistically analyse tasks (SA, NE, KW) independently from the individual systems, we observe that all three systems' types obtain a very low performance on the keyword extraction data sets (SemEval and Inspec). In terms of precision, SA are indistinguishable from NE (they obtain similar results) for the SA task but they outperform NE systems in recall. Finally, SA systems outperform NE systems for the NE task. This is not an expected result. Of course, named entities constitute a subset of all annotations, and thus should be detected by both SA and NE systems. But SA systems are trained to also detect other concepts that are not named entities. Thus, when applied on the MSNBC corpus, which contains only named entities, the hypothesis is that they should display lower values for precision, and it is not what we observe.

We also notice that NE systems do not return a lot of spots and fail to identify many relevant named entities. This might be explained by the emergence of several named entity types on the linked data cloud in contrast to the more traditional named entity detection task. Based on our experimental results, one key insight is that semantic annotators are better at annotating all concepts in documents rather than extracting a limited set of key phrases (keyword extraction), or relevant named entities, but are nevertheless the best performing systems for all three tasks.

There are some limitations to our study. One improvement to our work would be to report evaluation results based on a *semantic* approach. In our current work, we applied a *lexical* approach for matching key phrases and entities returned by semantic annotators with those indicated in the gold standards. As an example, based on this approach, two key-phrases 'parallel processes' and 'parallel processing' are matched because both have the same stem. This approach seems reasonable for the spotting phase. However, in the disambiguation part of the evaluation, this might cause some problems as two phrases will match only if both have been assigned to exactly the same entity in Wikipedia or any other knowledge base. This approach limits the evaluation performance by disregarding possible partial matches that may exist between key phrases. For example, given a gold standard phrase 'parallel processing method', systems that retrieve 'parallel processing', 'parallel systems' or nothing will be considered as equally unsuccessful. Cornolti et al. (2013b) attempt to address this issue by providing a *weak annotation match* mechanism. Based on this mechanism, two key phrases match if they overlap and refer to the same entity in the knowledge base. However, there are situations that cannot be handled by this partial matching approach. As an example, consider one gold standard key-phrase 'parallel computing', and two systems I and II that return 'parallel processing' and 'CPU' as key phrases, respectively. We can assume that both systems work better than a system that returns nothing for instance, while it is obvious that System I finds a closer match to the original keyword than System II. Cornolti et al. (2013b) try to address some of these concerns by applying Milne-Witten's relatedness measure between phrases (Medelyan et al., 2008). Nonetheless, their paper does not provide a set of formalized performance measures. To address these issues, a more comprehensive approach would have to deal with the semantics of links between entities on the linked open data cloud, such as sub-class, broader, narrower and similar links to evaluate the disambiguation performance.

Another important limit of our evaluation is due to the nature of the used data sets. It is well known that the

elaboration of a suitable data set for the evaluation of semantic annotators is a very difficult task. There are usually a large number of mentions that must be annotated in a document. This can hardly be achieved automatically, and thus results in data sets that contain few documents, and makes it difficult to obtain reliable statistics. Also, by inspecting the gold standards distributed with the data sets, we note that there are usually many incorrect and missing annotations (which we addressed only in MSNBC by re-annotating the data set). Finally, in our evaluation, we used only one or two data sets for each task, which may not be sufficient to avoid biases in the analysis. Still, we do think that our results provide some interesting hints on the behaviour of semantic annotators on the three tasks.

Acknowledgements

This work was partly funded by the Royal Military College of Canada Academic Research Program and sabbatical leave fund and the NSERC discovery grant program.

References

Andrade, M.A. and Valencia, A. (1998) 'Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families', *Bioinformatics*, Vol. 14, No. 7, pp.600–607.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R. and Hellmann, S. (2009) 'Dbpedia-a crystallization point for the web of data', *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 7, No. 3, pp.154–165.

Chen, L., Ortona, S., Orsi, G. and Benedikt, M. (2013) 'Aggregating semantic annotators', *Proceedings of the VLDB Endowment* Vol. 6, No. 13, pp.1486–1497.

Cornolti, M., Ferragina, P. and Ciaramita, M. (2013a) 'A framework for benchmarking entity-annotation systems', in *Proceedings of the 22nd International Conference on World Wide Web, International World Wide Web Conferences Steering Committee*, pp.249–260.

Cornolti, M., Ferragina, P. and Ciaramita, M. (2013b), 'A framework for benchmarking entity-annotation systems', *Proceedings of the 22nd International Conference on World Wide Web*, *International World Wide Web Conferences Steering Committee*, pp.249–260.

Cucerzan, S. (2007) 'Large-scale named entity disambiguation based on wikipedia data', *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp.708–716.

Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., Petrak, J. and Bontcheva, K. (2015) 'Analysis of named entity recognition and linking for tweets', *Information Processing and Management*, Vol. 51, No. 2, pp.32–49.

Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A. and Tomlin, J.A. et al. (2003), Semtag and Seeker: Bootstrapping the semantic web via automated semantic annotation, *in* 'Proceedings of the 12th international conference on World Wide Web', ACM, pp.178–186.

Ferragina, P. and Scaiella, U. (2010) 'Tagme: on-the-fly annotation of short text fragments (by wikipedia entities)', *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (CIKM'10), ACM, New York, NY, USA, pp.1625–1628.

Gangemi, A. (2013) 'A comparison of knowledge extraction tools for the semantic web', in *Extended Semantic Web Conference*, Springer, pp.351–366.

Jean-Louis, L., Zouaq, A., Gagnon, M. and Ensan, F. (2014) 'An assessment of online semantic annotators for the keyword extraction task', *Proceedings of the 13th Pacific Rim International Conference on Artificial Intelligence*, Gold Coast, Australia, pp.548–560.

Joksimovic, S., Jovanovic, J., Gasevic, D., Zouaq, A. and Jeremic, Z. (2013) 'An empirical evaluation of ontology-based semantic annotators', '*Proceedings of the 7th International Conference on Knowledge Capture*, ACM, pp.109–112.

Jonhson, R.A. and Wichern, D.W. (1992) *Applied Multivariate Statistical Analysis*, 3rd ed., Prentice-Hall.

Kim, S.N., Medelyan, O., Kan, M.-Y. and Baldwin, T. (2010) 'Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles', *Proceedings of the 5th International Workshop on Semantic Evaluation*, Association for Computational Linguistics, pp.21–26.

Kiryakov, A., Popov, B., Terziev, I., Manov, D. and Ognyanoff, D. (2011) 'Semantic annotation, indexing, and retrieval', *Web Semantics: Science, Services and Agents on the World Wide Web*, pp.1–16.

Kulkarni, S., Singh, A., Ramakrishnan, G. and Chakrabarti, S. (2009) 'Collective annotation of wikipedia entities in web text', *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp.457–466.

Medelyan, O., Witten, I.H. and Milne, D. (2008) Topic indexing with wikipedia, 'Proceedings of the AAAI WikiAI workshop', pp.19–24.

Meij, E. (2013*) A comparison of five semantic linkingalgorithms on tweets*. personal blog.

Mendes, P.N., Jakob, M., García-Silva, A. and Bizer, C. (2011) 'DBpedia Spotlight: shedding light on the web of documents', *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics, ACM, New York, NY, USA, pp.1–8.

Milne, D. and Witten, I.H. (2013) 'An open-source toolkit for mining wikipedia', *Artificial Intelligence*, Vol. 194, pp.222–239.

Moro, A., Raganato, A. and Navigli, R. (2014) 'Entity linking meets word sense disambiguation: a unified approach', *Transactions of the Association for Computational Linguistics*, Vol. 2, pp.231–244.

Nadeau, D. and Sekine, S. (2007) 'A survey of named entity recognition and classification', *Lingvisticae Investigationes*, Vol. 30, No. 1, pp.3–26.

Qureshi, M.A., O'Riordan, C. and Pasi, G. (2012) 'Short-text domain specific key terms/phrases extraction using an n-gram model with wikipedia', *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*', ACM, pp.2515–2518.

Ruiz, P. and Poibeau, T. (2015) 'Combining open source annotators for entity linking through weighted voting', *Joint Conference on Lexical and Computational Semantics (*SEM 2015)*', pp.211–215.

Shen, W., Wang, J. and Han, J. (2015) 'Entity linking with a knowledge base: Issues, techniques, and solutions', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 27, No. 2, pp.443–460.

Sinaci, A.A. and Gonul, S. (2012) 'Semantic content management with apache stanbol', *Extended Semantic Web Conference*', Springer, pp.371–375.

Usbeck, R., Röder, M., Ngonga Ngomo, A.-C., Baron, C., Both, A., Brümmer, M., Ceccarelli, D., Cornolti, M., Cherix, D., Eickmann, B. et al. (2015) 'Gerbil: general entity annotator benchmarking framework', *Proceedings of the 24th International Conference on World Wide Web*, pp.1133–1143.

Waitelonis, J., Jürges, H. and Sack, H. (2016) 'Don't compare apples to oranges: Extending gerbil for a fine grained nel evaluation', *Proceedings of the 12th International Conference on Semantic Systems*, ACM, pp.65–72.

Waitelonis, J., Jürges, H. and Sack, H. (2019) 'Remixing entity linking evaluation datasets for focused benchmarking', *Semantic Web*, Vol. 10, No. 2, pp.385–412.

Notes

1 The datasets and gold standards used in our evaluation are available http://www.labowest.ca/AnnotatorsEvaluation

2 https://www.ibm.com/watson/services/natural-language-understanding/

3 http://docs.aylien.com/

4 http://babelfy.org/

5 https://dandelion.eu/

6 http://www.opencalais.com/

7 https://github.com/dbpedia/dbpedia/wiki

8 http://tagme.di.unipi.it/

9 http://www.umbel.org/web-services/tagger-concept-noun/

10 https://developer.yahoo.com/contentanalysis/

11 https://www.textrazor.com/

12 http://ailab.ijs.si/tools/enrycher/

13 http://www.meaningcloud.com/

14 Note that we are using logistic regression as a statistical tool, not in a machine-learning perspective.