Supporting Natural Language Processing with Background Knowledge: Coreference Resolution Case

Volha Bryl, Claudio Giuliano, Luciano Serafini, and Kateryna Tymoshenko
{bryl,giuliano,serafini,tymoshenko}@fbk.eu

Fondazione Bruno Kessler, via Sommarive 18, 38123 Trento, Italy

Abstract. Systems based on statistical and machine learning methods have been shown to be extremely effective and scalable for the analysis of large amount of textual data. However, in the recent years, it becomes evident that one of the most important directions of improvement in natural language processing (NLP) tasks, like word sense disambiguation, coreference resolution, relation extraction, and other tasks related to knowledge extraction, is by exploiting semantics. While in the past, the unavailability of rich and complete semantic descriptions constituted a serious limitation of their applicability, nowadays, the Semantic Web made available a large amount of logically encoded information (e.g. ontologies, RDF(S)-data, linked data, etc.), which constitutes a valuable source of semantics. However, web semantics cannot be easily plugged into machine learning systems. Therefore the objective of this paper is to define a reference methodology for combining semantic information available in the web under the form of logical theories, with statistical methods for NLP. The major problems that we have to solve to implement our methodology concern (i) the selection of the correct and minimal knowledge among the large amount available in the web, (ii) the representation of uncertain knowledge, and (iii) the resolution and the encoding of the rules that combine knowledge retrieved from Semantic Web sources with semantics in the text. In order to evaluate the appropriateness of our approach, we present an application of the methodology to the problem of intra-document coreference resolution, and we show by means of some experiments on the standard dataset, how the injection of knowledge leads to the improvement of this task performance.

1 Introduction

The two key aspects of natural language applications based on machine learning techniques are the learning algorithm, and the feature extraction and representation of the documents, entities, or words that have to be manipulated. Reviewing the relevant literature of the last years, one realizes that, typically, the difference between the results obtained by different learning algorithms (e.g., support vector machines vs. decision trees) is significant when they are fed with the same information. On the other hand, the feature extraction and representation methods play a crucial role for the accuracy of the system. Simple representations, e.g., the bag-of-words, and more complex ones, e.g., tree kernels, have been exploited in different tasks and their difference has been proved to be significant as well. For example, in relation extraction approaches that exploit deep syntactic parsing outperform the ones that represent only shallow syntactic analysis. Until now, the majority of the approaches focus on representing syntactic information while background knowledge extracted from knowledge bases has been restricted to WordNet and ad-hoc gazetteers [12, 7]. The main reasons are due to the low coverage of the available knowledge resources and the difficulty to match text and ontology elements.

Nowadays, the Semantic Web made available a large amount of logically encoded information (e.g., ontologies, RDF(S)-data, linked data, etc.), which constitute a valuable source of semantic knowledge. However, extending the state-of-the-art natural language applications to use these resources is not a trivial task due to the following reasons: (i) The *heterogeneity* and the *ambiguity* of the schemes adopted by the different resources of the Semantic Web. This means, e.g., that the same relation can be encoded by different URIs, and that URIs are used by different resources for denoting different relations. (ii) The *irregular coverage* of the knowledge available in the Web. This means that for some "famous" entities the Semantic Web contains a large amount of knowledge, and only a little is relevant for solving a specific task (e.g., coreference resolution or relation extraction), while for other entities there is no knowledge at all. (ii) The *logical-statistical knowledge integration problem*, i.e., the fact that algorithms for coreference resolution are based on statistical feature models, while background knowledge in the Semantic Web is encoded in some logical form.

In this paper, we define a general methodology for supporting natural language processing by exploiting background knowledge available in the Web, by proposing practical solutions for the before mentioned problems. *First*, we map terms in text to URIs through Wikipedia mediation. Since most of the resources available in the Semantic Web are linked to Wikipedia, we can use it as a *semantic mediator*. So we propose to link text with Wikipedia entries and then to exploit the linking between Wikipedia and the other resources to access the knowledge encoded in them. Wikipedia represents a practical choice, as it is playing a central role in the development of the Semantic Web, given the large and growing number of resources linked to it, which makes Wikipedia one of the central interlinking hubs of the emerging Web of Data. *Second*, we query the Semantic Web using the URIs to obtain the background knowledge expressed in the RDF/OWL formalism and apply feature selection techniques to retrieve the relevant knowledge for the specific task. In this way we do not assume to have any a priori knowledge of the specific task but we delegate to the *feature selection* phase the responsibility of finding the relevant information from an arbitrary Semantic Web resource to model it. Differently, in our previous work [5] we experimented with the small predefined subset of properties from one specific knowledge source (YAGO ontology) to support the coreference resolution task. *Finally*, as we presented in more details in [5], we use the Alchemy tool [1] for the integration of uncertain knowledge, and facts expressed in first-order language. Alchemy provides both reasoning and learning functionalities, though we only use the reasoning part. The extension of this work, however, could require learning capabilities.

To evaluate the methodology, we run a number of experiments in coreference resolution, which are reported in Section 5. The experiments consist in selecting a set of features relevant for the given task from three large-scale Semantic Web resources and then testing the coreference resolution model extended with the selected features. The results show that our method performs in the order of the state-of-the-art coreference algorithms, and, importantly, that the use of background knowledge provides a tangible advantage for coreference resolution.

2 Coreference Resolution: Task Definition and Related Work

The task of coreference resolution consists in identifying mentions that refer to the same real-world entity. E.g., it is required to identify that the mentions *Barack Obama* and *president* are coreferent in the text *"Barack Obama will make an appearance on the TV show. The president is scheduled to come on Friday evening."* This constitutes an important subtask in many natural language processing (NLP) applications such as information extraction, textual entailment, and question answering. Machine learning (ML) is widely used to approach the coreference task. State-of-the-art coreference resolvers are mostly extensions of the Soon et al. approach in which a mention-pair classifier is trained using solely surface-level features to determine whether two mentions are coreferring or not [25]. In the last decade, two independent research lines have extended the Soon et al. approach yielding significant improvements in accuracy.

The first aims at defining a more sophisticated ML framework to overcome the limits of the mention-pair model. Entity-mention and mention-ranking models and their combination cluster-ranking are some of the relevant approaches proposed (e.g. [9, 16]). An entity-mention model considers candidate pairs, which consist of a cluster of mentions, referring to the same entity, and a new mention. [18] motivate the entity-mention model using an example of a mention set such as "Mr. Obama", "Obama" and "she". A mention-pair model might first predict that "Mr. Obama" and "Obama" are coreferent, then it might predict that "Obama" and "she" are coreferent as well, and finally cluster all these mentions as referring to the same entity. An entity-mention model first classifies "Mr. Obama" and "Obama" as coreferent, and then immediately clusters them into an entity cluster. Then the model considers the *entity cluster* ("Mr. Obama", "Obama") and the mention "she" as the coreference candidates. In this case "she" is unlikely to be added to the given cluster, as there is a gender disagreement between "Mr." and "she". The mention-ranking models attempt to choose the most probable candidate antecedent for a mention, among *all* the preceding mentions within a given scope. E.g., if a text contains the mentions "she", "Barack Obama" and "Michele Obama", the set of candidate antecedents for the mention ''Michele Obama" includes ''she" and ''Barack Obama". The models ranks them and chooses the most probable one.

The second research line investigates the usage of semantic knowledge sources to augment the feature space [25, 20, 17, 27]. Here the majority of the approaches exploit WordNet [11] and, more recently, Wikipedia[1] or corpora annotated with semantic classes. E.g., in [25] a candidate pair of mentions was represented as a vector of twelve features, two of which, namely the semantic class agreement and alias, were of semantic nature. The alias feature contributed greatly to the performance of the system. It was obtained using a set of heuristics, e.g. it was considered true if one mention was an

[1] `http://wikipedia.org/`

acronym of another. Therefore, its value could be evaluated only in a limited number of cases. The semantic class agreement feature did not impact the performance of the system, which may be due to the fact that the most frequent sense of a mention in the WordNet lexical database was employed as its semantic class. Therefore, the possible ambiguity of a mention was not taken into account. In [19] a set of features from [25] was expanded, with the semantic relatedness features based on WordNet taxonomy. However, they did not perform the disambiguation as well, and the new semantic features did not impact the final performance of the system either. Recently, Wikipedia has also started to be exploited as a source of semantic knowledge for coreference resolution [20, 27]. E.g., its category structure and article texts are used in [20] in order to obtain a set of six features based on the semantic relatedness of mentions. In order to find the Wikipedia articles which correspond to a mention, Wikipedia is queried for pages titled as the head lemma of the mention. If the disambiguation page is hit, an heuristic algorithm is employed. However, such approach is likely to return the Wikipedia page that corresponds to the most frequent sense of a mention. The problem of possible noun mention ambiguity was taken into account in [17]. In this work a special classifier was trained on the BBN entity corpus to assign one of five semantic classes to the mentions. Even though the set of semantic classes is not large, the features based on usage of these classes gave an improvement of the precision of the common noun resolution by 2-6% over [25]. These results show that taking into account the ambiguity of the mentions is crucial for obtaining the semantic knowledge relevant for coreference resolution. Knowledge representation format and the structure of the knowledge sources used by the above described approaches are different, therefore, in each specific case information from a resource has to be extracted and processed differently. In the following section we present an approach that allows us to overcome this issue and work with knowledge from heterogeneous sources with only minimal assumptions on their representation and structure.

3 Background Knowledge Acquisition

3.1 Sources of Background Knowledge

Our approach is concerned with using background knowledge from multiple resources in a unified way. We propose to acquire it from collections of RDF data, made available by the members of the Linked Data Community, e.g., DBpedia [2], Freebase [4], YAGO [26], and, perspectively, many others. In order to obtain semantic knowledge about a mention in plain text, we need to map it to a Linked Data resource entry. We benefit from the fact that some of the Linked Data resources are aligned with Wikipedia. Therefore, we link a mention to Wikipedia, using an approach described in Section 3.2, and then exploit this link to obtain data from the specific Linked Data resource. Moreover, Linked Data datasets are interconnected by means of RDF links and in future these inter-dataset links can be exploited as well. In the current work, we limit the scope of our research to the following resources, that can be directly accessed by using a Wikipedia link:

DBpedia is a structured twin of Wikipedia. Currently it describes more than 3.4 million entities. DBpedia resources bear the names of the Wikipedia pages, from which they have been extracted.

YAGO is an automatically created ontology, with taxonomy structure derived from WordNet, and knowledge about individuals extracted from Wikipedia. Therefore, the identifiers of resources describing individuals in YAGO are named as the corresponding Wikipedia pages. YAGO contains knowledge about more than 2 million entities and 20 million facts about them.

Freebase is a collaboratively constructed database. It contains knowledge automatically extracted from a number of resources including Wikipedia, MusicBrainz,[2] and NNDB,[3] as well as the knowledge contributed by the human volunteers. Freebase describes more than 12 million interconnected entities. Each Freebase entity is assigned a set of human-readable unique keys, which are assembled of a value and a namespace. One of the namespaces is the Wikipedia namespace, in which a value is the name of the Wikipedia page describing an entity.

3.2 Linking to Wikipedia

The linking problem is cast as a word sense disambiguation (WSD) exercise, in which each mention in the text (excluding pronouns) has to be disambiguated using Wikipedia to provide the sense inventory and the training data. The idea of using Wikipedia to train a supervised WSD system was first proposed in [6]. The proposed approach, called *The Wiki Machine*,[4] is summarized as follows.

Training Set To create the training set, for each mention m, we collect from the English Wikipedia dump[5] all contexts where m is an anchor of an internal link, where a context corresponds to a line of text in the Wikipedia dump and it is represented as a paragraph in a Wikipedia article. The set of target articles represents the senses of m in Wikipedia and the contexts are used as labeled training examples. E.g., the proper noun *Bush* is a link anchor in $17,067$ different contexts that point to 20 different Wikipedia pages, `George_W._Bush`, `Bush_(band)`, and `Dave_Bush` are some example of possible senses. The set of contexts with their corresponding senses is then used to train the WSD system described below. E.g., the context "*Alternative Rock bands from the mid-90 's , including Bush , Silverchair , and Sponge.*" is a training instance for the sense defined by the Wikipedia entry `Bush_(band)`.

Learning Algorithm To disambiguate mentions in text, we implemented a kernel-based approach originally proposed in [13]. Different kernel functions are employed to integrate syntactic, semantic, and pragmatic knowledge sources typically used in the WSD literature. Kernel methods are theoretically well founded in statistical learning theory and shown good empirical results in many applications [24]. The strategy

[2] http://musicbrainz.org/

[3] http://www.nndb.com/

[4] http://thewikimachine.fbk.eu/

[5] http://download.wikimedia.org/enwiki/20100312

adopted consists in splitting the learning problem into two parts. They first embed the input data in a suitable feature space, and then use a linear algorithm (e.g., support vector machines) to discover nonlinear patterns in the input space. The kernel function is the only task-specific component of the learning algorithm. For each knowledge source a specific kernel has been defined. By exploiting the property of kernels, basic kernels are then combined to define the WSD kernel. Specifically, we used a linear combination of gap-weighted subsequences, bag-of-words, and latent semantic kernels .

Gap-weighted subsequences kernel This kernel learns syntactic and associative relations between words in a local context. We extended the gap-weighted subsequences kernel to subsequences of word forms, stems, part-of-speech tags, and orthographic features (capitalization, punctuation, numerals, etc.). We defined gap-weighted subsequences kernels to work on subsequences of length up to 5. E.g., suppose we have to disambiguate the verb to score in the context “Maradona scored Argentina’s third goal”, given the labeled example “Ronaldo scored two goals in the second half” as training, a traditional approach, that only consider contiguous ngrams, has no clues to return the correct answer because the two contexts have no features in common. The use of gap-weighted subsequences allows us to overcame this problem and extract the feature “score goal”, shared by the two examples.

Bag-of-words kernel This kernel learns domain, semantic, topical information. Bag-of-words kernel takes as input a a wide context window around the target mention. Words are represented using stems. The main drawback of this approach is the need of a large amount of training data to reliably estimate model parameters. E.g., despite the fact that the examples “People affected by AIDS” and “HIV is a virus” express concepts related, their similarity is zero using the bag-of-words model since they have no words in common (they are represented by orthogonal vectors). On the other hand, due to the ambiguity of the word virus, the similarity between the contexts “the laptop has been infected by a virus” and “HIV is a virus is greater than zero”, even though they convey very different messages.

Latent semantic kernel To overcome the drawback of the bag-of-words, we incorporate semantic information acquired from English Wikipedia in an unsupervised way by means of latent semantic kernel. This kernel extracts semantic information through co-occurrence analysis in the corpus. The technique used to extract the co-occurrence statistics relies on a singular value decomposition of the term-by-document matrix. E.g., the similarity in the latent semantic space of the two examples “People affected by AIDS” and “HIV is a virus” is higher than in the bag-of-words representation, because the terms AIDS, HIV and virus very often co-occur in the medicine domain.

Implementation Details The latent semantic model is derived from the 200,000 most visited Wikipedia articles. After removing terms that occur less than 5 times, the resulting dictionary contain about 300,000 and 150,000 terms respectively. We used the SVDLIBC package to compute the SVD, truncated to 400 dimensions.[6] To classify each mention in Wikipedia entries, we used a LIBSVM package.[7] No parameter optimization was performed.

[6] `http://tedlab.mit.edu/~dr/svdlibc/`

[7] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

Evaluation We evaluate The Wiki Machine on the ACE05-WIKI Extension [3]. This dataset extends the the English Automatic Content Extraction (ACE) 2005 dataset with ground-truth links to Wikipedia.[8] ACE 2005 is composed of 599 articles assembled from a variety of sources selected from broadcast news programs, newspapers, newswire reports, internet sources and from transcribed audio. It contains the annotation of a series of entities (person, location, organization) and their mentions. In the extension each nominal or named entity mention (in total 29,300 entity mentions) is manually assigned a Wikipedia link(s). The results of the evaluation are reported in the first line of Table 1. The training sets were collected from the English Wikipedia dump of March, 2010.

We have compared our approach with the state-of-the-art system described in [15]. In this approach, a plain text is *wikified*, i.e. terms in the text are linked to Wikipedia and then keywords are selected among them. We are interested only in the linking step. In this step a set of candidate Wikipedia pages (senses) for all terms in the text is collected as described in Section 3.2, when possible. The pages to which terms in the text can be linked unambiguously form the *context*. Different senses of an ambiguous term are evaluated using a classifier, based on three features, namely *commoness* of a sense, its *relatedness* to the context and the *context quality*.

The approach is implemented in the *Wikipedia Miner* tool.[9] We used it with the default parameters. The tool requires a Wikipedia dump preprocessed in a special way. We used the preprocessed Wikipedia dump of July, 2008, made available by the authors of the tool. The results are reported in the second line of Table 1. The Wikipedia Miner achieves six points better precision, however, its recall is considerably lower, thus making the F_1 13 points less than that of The Wiki Machine. The performance difference between the two systems could not be only due to the use of different version of Wikipedia, as the ACE corpus contains references to entities dated before 2005 and Wikipedia covered most of them in 2008. On the other hand, varying the Wiki Miner free parameters did not produce significant improvement.

Approach	**Precision**	**Recall**	$\mathbf{F_1}$
The Wiki Machine	0.716	0.714	0.715
Wikipedia Miner	0.779	0.471	0.587

Table 1. Comparisons of the two linking methods on the ACE05-WIKI Extension.

4 Selecting Relevant Background Knowledge

The amount of information obtained from a Semantic Web resource even for a single named entity can be very big. For instance, DBpedia alone contains around 600 RDF triples describing *Barack Obama*. Most of this information is irrelevant to the NLP

[8] `http://www.itl.nist.gov/iad/mig//tests/ace/ace05/index.html`

[9] `http://wikipedia-miner.sourceforge.net/`

task at hand (e.g. Obama's website, residence, the name of his spouse, etc.), and only some of the triples can be useful to resolve coreferences (e.g. *type* properties stating that Obama is a politician and a president).

Indeed, many learning algorithms are originally not designed to deal with large amounts of irrelevant information, consequently, combining them with the *feature selection* techniques has become necessary in many applications. This is particularly true when the information needed is retrieved from heterogeneous knowledge sources as the ones made available on the Semantic Web. Recall that we do not assume any prior knowledge on the nature of the background knowledge that can be obtained, barring the availability in RDF.

We use the chi-square test to assess the relevance of background knowledge for the coreference resolution task by looking only at the intrinsic properties of the data. The chi-square test is a test for dependence between a feature and a class. Specifically, chi-square metric is calculated for each feature, and low-scoring features are removed. Afterwards, this subset of features is presented as input to the learning algorithm. Benefits of the chi-square test are that it easily scales to very high-dimensional data sets, it is computationally simple and fast, and the search in the feature space is separated from the search in the hypothesis space. The next sections describe the feature extraction and selection methods.

4.1 Feature Extraction

We obtain feature sets for coreference candidates, in which mentions are either a proper noun and a common noun (NAM-NOM), or both are common nouns (NOM-NOM). We denote a coreference candidate pair by (m_1, m_2). In the case of a NAM-NOM pair m_1 refers to the proper noun mention and m_2 to the common noun mention. As regards NOM-NOM, we consider two (m_1, m_2), pairs which differ by the order of the mentions, e.g. for the coreference candidate ("state", "country") we consider (m_1="state", m_2 = "country") and (m_1 = "country", m_2 = "state").

An (m_1, m_2) pair is processed as follows. We extract all RDF triples referring to m_1 from a knowledge source, using the methodology described in Section 3. In average we obtain 200 triples per mention. An RDF triple consists of subject, predicate and object. If m_1 is the object of the triple, we check if there is a string match between m_2 and the subject. In the other case, we check whether there is a string match between m_2 and the object. If the string match is observed, then the coreference candidate pair has a feature named as the predicate of the RDF triple, and the feature is included into the feature set. If for RDF-triples with a given predicate the string match never occurs in the entire training set, then the corresponding feature is not included into the feature set.

Examples of features for some of the mention pairs are presented in Table 2. Each mention is composed of the number of a document, the position in the document and the mention string itself. We select distinct sets of features for NAM-NOM and NOM-NOM mentions of person (PER) and geopolitical entities (GPE). Consequently from each of three background knowledge sources we extract four sets of features, namely NAM-NOM-GPE, NOM-NOM-GPE, NAM-NOM-PER, and NOM-NOM-PER. They typically contain 10-50 features. We apply the feature selection technique to each set.

Mention pair	**Feature**
1-225-Clinton, 1-87-president	http://www.w3.org/2004/02/skos/core#subject
529-324-Yasser Arafat, 529-402-leader	http://www.w3.org/2004/02/skos/core#subject
410-23-state, 410-109-country	http://www.w3.org/2004/02/skos/core#subject
2-637-Kuwait, 2-956-city	http://rdf.Freebase.com/ns/location.country.capital
3-10-U.S.,3-892-States	http://www.w3.org/2002/07/owl#sameAs

Table 2. Feature examples

4.2 Feature Selection

In machine learning coreference candidates are called instances. We say than an instance belongs to class 1 if the mentions in the candidate pair are coreferent; 0 otherwise. Let us introduce some notation.

n_{1f} number of instances in class 1 with feature *f*
$n_{1\bar{f}}$ number of instances in class 1 without feature *f*
n_{0f} number of instances in class 0 with feature *f*
$n_{0\bar{f}}$ number of instances in class 0 without feature *f*
n_1 total number of instances in class 1
n_0 total number of instances in class 0
n_f total number of instances with feature *f*
$n_{\bar{f}}$ total number of instances without feature *f*
n total number of instances

The chi-square feature selection metric, $\chi^2(f, c)$, measures the dependence between feature f and class $c \in \{0, 1\}$. If f and c are independent, then $\chi^2(f, c)$ is equal to zero. To select a relevant set of features, we utilized the following metric

$$\chi^2(f,c) = \frac{n(n_{1f}n_{0\bar{f}} - n_{0f}n_{1\bar{f}})^2}{n_1 n_f n_0 n_{\bar{f}}},$$

by averaging over the classes we obtain the metric for selecting a subset of features

$$\chi^2(f) = \sum_{i=0}^{1} Pr(c_i)\chi^2(f,c).$$

E.g., we extract from Freebase a set of 22 features for the NAM-NOM pairs of mentions which refer to a GPE entity. After feature selection, the scores of 9 features are near to zero, consequently only 13 features should be considered. The two top-scoring features in this case are `http://www.w3.org/2002/07/owl#sameAs` and `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`. These features and their equivalents in other knowledge sources turned out to be highly relevant for other kinds of coreference as well.

5 Evaluation: Coreference Resolution with Background Knowledge

In this section we report on our experiments with the coreference resolution task. Namely, we give some hints on the implementation of the model we used as a baseline (more details can be found in [5]), explain how the background knowledge is plugged into the model, and present the results of the experiments.

5.1 Baseline Model Definition

Tool Selection A recently introduced family of approaches to the task of coreference resolution try to represent the coreference task into some logical theory that supports the representation of uncertain knowledge. Among these approaches we can find a number of works [22, 14, 8] based on the formalism called Markov logic [10], which is a first-order probabilistic language which combines first-order logic with probabilistic graphical models.

In essence, Markov logic model is a set of first-order rules with weights associated to each rule. Weights can be learned from the available evidence (training data) or otherwise defined, and then inference is performed on a new (test) data. Such a representation of the model is intuitive and allows for the background knowledge be integrated naturally into it. It has been shown that the Markov logic framework is competitive in solving NLP tasks (see, for instance, [21, 23], and [1] for more references). Another advantage of the weighted first-order representation is that the model can be easily extended with extra knowledge by simply adding logical axioms, thus minimizing the engineering effort and making the knowledge enrichment step more straightforward and intuitive.

Given the above, the inference tool we have selected to be used in the coreference resolution tasks is the inference module of the Alchemy system [1], with Markov logic as a representation language.

The Alchemy inference module takes as inputs (i) a Markov logic model, that is, a list of weighted first-order rules, and (ii) an evidence database, that is, the list of known properties (true of false values of predicates) of domain objects. In the case of coreference resolution, domain objects are the entity mentions, and the properties they might have are gender, number, distance, semantic class, etc. In the following we discuss how these two parts of input are constructed.

Markov Logic Model In defining a model for coreference resolution, we were inspired by Soon et al baseline [25], which uses the following features: pairwise distance (in terms of number of sentences), string match, alias, number, gender and semantic class agreement, pronoun, definite/demonstrative noun phrase and both proper names feature. This approach achieves an F-measure of 62.2% in the MUC-6 coreference task and of 60.4% on the MUC-7 coreference task.

A Markov logic model consists of a list of predicates and a set of weighted first-order formulae. Some predicates in our model correspond to Soon et al features: binary predicates such as *distance* between two entity mentions (in terms of sentences) and string match, and unary predicates such as *proper name*, *semantic class*, number (*singular* or *plural*) and gender (*male*, *female* or *unknown*). Also, we use *string overlap* in addition to *string match* and define yet another predicate to describe *distance*, which refers to the number of named entities of the same type between two given ones (e.g. if there are no other named entities classified as “person” between “Obama” and “President”, the distance is 0). The predicate *corefer(mention,mention)* describes the relation of interest, and is called *query* predicate in Alchemy terminology, that is, we are interested in evaluating the probability of each grounding of this predicate given the known properties of all the mentions.

The second part of the model definition concerns constructing the first-order rules appropriate for a given task. We have defined the rules that connect the above properties of the mentions with the coreference property. Some of the examples are given below[10].

String match is very likely to indicate coreference for proper names, while for common nouns it is still likely but makes more sense in combination with a distance property:

$$20\ match(x,y) \wedge proper(x) \wedge proper(y) \rightarrow corefer(x,y)$$

$$3\ match(x,y) \wedge noun(x) \wedge noun(y) \wedge dist0(x,y) \rightarrow corefer(x,y)$$

The number before a formula corresponds to the *weight* assigned to it.

Gender and number agreement between two neighboring mentions of the same type provides a relatively strong evidence for coreference:

$$4\ male(x) \wedge male(y) \wedge singular(x) \wedge singular(y) \wedge follow(x,y) \rightarrow corefer(x,y)$$

We also define hard constraints, that is, crisp first-order formulae that should hold in any given world. Fullstop after the formula refers to an infinite weight, which, in turn, means that the formula holds with the probability equal to 1.

$$\neg corefer(x,x).$$

$$corefer(x,y) \wedge \rightarrow corefer(y,x).$$

In this paper we do not consider weight learning, so weights are assigned manually. We do not consider pronoun mentions as the background knowledge is relevant for proper name/common noun pairs in the first place.

Evidence Database The second input to the Alchemy inference module is an evidence database, i.e. the known values of non-query predicates listed in the previous section. Normally, the coreference resolution task is performed on a document corpus, in which each document is firstly preprocessed. Preprocessing consists in identifying the named entities (persons, locations, organization, etc.), as well as their syntactic properties, such as part of speech, number, gender, pairwise distance, etc.

The data corpus we use for the experiments is ACE 2005 data set, with around 600 documents from the news domain. We work on a corpus in which each word is annotated with around 40 features (token and document ID, Part of Speech tags by TextPro[11], etc.). This allowed us to extract the syntactic properties of the mentions presented before. Note that for the gender property, we used male/female name lists to annotate proper names in the corpus. For common nouns, we defined two lists of gender tokens (which included “man”,“girl”, “wife”, “Mr.”, etc.). Some examples of the properties we obtained are given below.

[10] Full model is available at `https://copilosk.fbk.eu/images/1/1f/Coreference2.txt`

[11] TextPro – `http://textpro.fbk.eu/`

semclass (“2-83-Bob Dornan”, “person”)
neihgbourNouns (“2-82-Congressman”,“2-83-Bob Dornan”)
propername (“2-83-Bob Dornan”)
male (“2-83-Bob Dornan”)
singular (“2-83-Bob Dornan”)
pmatch (“2-740-Bob”, “2-83-Bob Dornan”)
match (“2-83-Bob Dornan”, “2-942-Bob Dornan”)
DBPedia_NAM-NOM_PER_2_type (“2-83-Bob Dornan”, “2-62-Congressman”)
YAGO_NAM-NOM_PER_1_type (“2-83-Bob Dornan”, “2-86-Republican”)

We worked on the gold standard annotation for named entities, and considered five named entity types: PERson, LOCation, GeoPoliticalEntity, FACility and ORGanization (although only the first two types were used in the experiments presented later in this section). Alchemy inference was performed separately for each named entity type. Note that the size of the document corpus does not impact the quality of the results as documents are processed independently, one by one.

The Alchemy inference module, which takes as input the weighted Markov logic model and the database containing the properties of mentions, produces as a result the probabilities of coreference for each of NxN possible pairs of mentions, where N is the number of mentions:

$$corefer(m_i, m_j) \quad p_{ij}, \quad 0 \leq p_{ij} \leq 1,\ i, j = \overline{1, N}$$

After having obtained this, we setup a probability threshold (e.g. $p = 0.9$) and consider only those pairs for which $p_{ij} \geq p$. On these pairs, we perform a transitive closure. Then the pairwise scores and, after a simple clustering step, MUC scores [28] are calculated. The resulting output consists of the list of coreference chains for each of the processed documents, and the measures of the efficiency, namely, recall, precision and their harmonic mean (F1).

5.2 Injecting Background Knowledge into Coreference Model

In the Markov logic model, in addition to the syntactic predicates and rules described above, a set of predicates and rules that deal with background knowledge were introduced. The predicates, or pairwise semantic properties of mentions, are the most relevant features selected according to the methodology described in Section 4 from the DBpedia, YAGO and Freebase knowledge sources. The list of the selected features is given in Table 3.

The Markov logic model is extended with the rules relating these semantic predicates with the coreference property. The arguments of a semantic predicate should be of the same named entity type (person or geopolicical entity), and the distance relation relation must hold between them.

For the experiments, the ACE data set was first ordered by the number of named entities linked to Wikipedia and split into two subsets of equal size (*ACE-SUBSET-1* and *ACE-SUBSET-2*): odd documents from the ordered list formed the first subset, even formed the second one. *ACE-SUBSET-1* was used for feature selection, while on *ACE-SUBSET-2* the Markov logic model extended with background knowledge was

KB name	NE type	Pair type	Property name
Freebase	GPE	NAM-NOM	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
Freebase	GPE	NAM-NOM	http://www.w3.org/2002/07/owl#sameAs
Freebase	PER	NAM-NOM	http://www.w3.org/2002/07/owl#sameAs
Freebase	PER	NAM-NOM	http://rdf.freebase.com/ns/people.person.profession
Freebase	PER	NOM-NOM	http://www.w3.org/2002/07/owl#sameAs
YAGO	GPE	NAM-NOM	type
YAGO	GPE	NAM-NAM	means
YAGO	PER	NAM-NOM	type
DBPedia	GPE	NAM-NOM	http://dbpedia.org/property/reference
DBPedia	GPE	NAM-NOM	http://www.w3.org/2004/02/skos/core#subject
DBPedia	GPE	NAM-NOM	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
DBPedia	PER	NAM-NOM	http://www.w3.org/2004/02/skos/core#subject
DBPedia	PER	NAM-NOM	http://www.w3.org/1999/02/22-rdf-syntax-ns#type
DBPedia	PER	NAM-NOM	http://dbpedia.org/property/title

Table 3. Selected features

tested. For the latter experiments, we have created yet another document set, *ACE-SUBSET-3*, which contains 50 documents from *ACE-SUBSET-2* with the highest background knowledge coverage (i.e. with the highest number of entity mentions linked to Wikipedia).

Tables 4 and 5 present MUC scores of the experiments for *ACE-SUBSET-2* and *ACE-SUBSET-3*, accordingly. Each table reports the values of MUC recall, precision and F1 for the models without and with the use of background knowledge extracted from DBpedia, YAGO and Freebase. Experiments were conducted for geopolitical entities (GPE) and persons (PER). Compared to the other three NE types (locations, organizations and facilities), persons and geopolitical entities constitute the major part of the corpus, so we do not report these results here. Also, we do not report the experiments for geopolitical entities with knowledge obtained from Freebase and DBpedia as the corresponding improvement for these cases was insignificant.

NE type	KB	R	P	F1
GPE	none	0.7446	0.9371	0.8298
GPE	YAGO	0.8314	0.9308	**0.8783**
PER	none	0.7003	0.7302	0.7149
PER	DBpedia	0.7125	0.7196	0.7160
PER	Freebase	0.7178	0.7343	**0.7259**
PER	YAGO	0.7208	0.7348	**0.7277**

Table 4. MUC scores for GPE and PER NE types, *ACE-SUBSET-2* document set

NE type	KB	R	P	F1
GPE	none	0.7763	0.9380	0.8495
GPE	YAGO	0.8536	0.9335	**0.8918**
PER	none	0.7447	0.6946	0.7188
PER	DBpedia	0.7669	0.6852	**0.7238**
PER	Freebase	0.7749	0.7024	**0.7369**
PER	YAGO	0.7785	0.7039	**0.7393**

Table 5. MUC scores for GPE and PER NE types, *ACE-SUBSET-3* document set

The improvement in $F1$ is 5% for GPE due to the use of YAGO on both datasets. The improvement in $F1$ for PER with the use of YAGO and Freebase is a bit higher for *ACE-SUBSET-3* (1.5% versus 2%) due to the increase of coverage in the latter. The results for YAGO and Freebase are comparable to the ones presented in [5], while lower improvement for DBpedia is most probably due to the fact that this knowledge source is much less structured and polished with respect to YAGO and Freebase.

6 Conclusion and Future Work

In this paper we have defined a methodology for supporting a natural language processing task with semantic information available in the Web under the form of logical theories. In order to empower an NLP task with the knowledge from publicly available large scale knowledge sources, we map the terms in the text to concepts in Wikipedia and then, to other knowledge resources linked to Wikipedia (DBpedia, Freebase and YAGO). An important aspect of the mapping that was addressed in the paper is word sense disambiguation. We have applied the proposed approach to the task of intra-document coreference resolution. We have proposed a method for selecting a subset of knowledge relevant for a given text for solving the coreference task, which is based on feature selection algorithms. We have implemented the coreference resolution process with the help of the inference module of the Alchemy tool. The latter is based on Markov logic formalism and allows combining logical and statistical representation and inference. The results were evaluated on the ACE 2005 data set.

To the best of our knowledge, there are no approaches nor to coreference resolution, neither to other NLP tasks, which make use of structured semantic knowledge available in the Web. One of the key points in addressing this problem is combining the logic based representation of the model with statistical reasoning. Such model representation and the available Semantic Web knowledge resources "speak the same language", which is the language of logic. Another important point of our approach is that no prior assumptions on the structure of the Semantic Web knowledge sources are needed for them to be used to support an NLP task.

Future work directions include further exploiting the Linked Data resources (including the one not used in this paper, e.g. Cyc[12]) to extract more properties and rules

[12] `http://www.cyc.com`

to support coreference resolution, as well as using the links between different Linked Data resources to obtain more knowledge. Also, we are interested in experimenting with the full task, which includes named entity recognition module and learning the weights of the formulae of the model from the training data. Testing the proposed reference methodology on the other NLP task, like semantic relation extraction, is another challenging future work direction.

Acknowledgments

The research leading to these results has received funding from the ITCH project (http://itch.fbk.eu), sponsored by the Italian Ministry of University and Research and by the Autonomous Province of Trento, and the Copilosk project (http://copilosk.fbk.eu), a Joint Research Project under Future Internet – Internet of Content program of the Information Technology Center, Fondazione Bruno Kessler.

References

1. Alchemy – http://alchemy.cs.washington.edu/
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: Dbpedia: A nucleus for a web of open data. In: ISWC/ASWC. pp. 722–735 (2007)
3. Bentivogli, L., Forner, P., Giuliano, C., Marchetti, A., Pianta, E., Tymoshenko, K.: Extending English ACE 2005 Corpus Annotation with Ground-truth Links to Wikipedia. In: 23rd International Conference on Computational Linguistics. pp. 19–26 (2010)
4. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250. ACM (2008)
5. Bryl, V., Giuliano, C., Serafini, L., Tymoshenko, K.: Using background knowledge to support coreference resolution. In: 19th European Conference on Artificial Intelligence (ECAI 2010). pp. 759–764 (2010)
6. Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). pp. 708–716 (2007)
7. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. pp. 423–431. Association for Computational Linguistics (2004)
8. Culotta, A., Wick, M.L., McCallum, A.: First-order probabilistic models for coreference resolution. In: Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. pp. 81–88 (2007)
9. Denis, P., Baldridge, J.: Joint determination of anaphoricity and coreference resolution using integer programming. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics. pp. 236–243 (2007), http://www.aclweb.org/anthology/N/N07/N07-1030
10. Domingos, P., Kok, S., Lowd, D., Poon, H., Richardson, M., Singla, P.: Markov logic. In: Probabilistic Inductive Logic Programming. Lecture Notes in Computer Science, vol. 4911, pp. 92–117. Springer (2008)
11. Fellbaum, C., et al.: WordNet: An electronic lexical database. MIT press Cambridge, MA (1998)

12. Giuliano, C., Lavelli, A., Pighin, D., Romano, L.: FBK-IRST: Kernel methods for semantic relation extraction. In: Proceedings of the 4th International Workshop on Semantic Evaluations. pp. 141–144. Association for Computational Linguistics (2007)
13. Giuliano, C., Gliozzo, A.M., Strapparava, C.: Kernel methods for minimally supervised wsd. Computational Linguistics 35(4), 513–528 (2009)
14. Huang, S., Zhang, Y., Zhou, J., Chen, J.: Coreference resolution using Markov Logic Networks. In: Proceedings of CICLing. pp. 157–168 (2009)
15. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: CIKM '08: Proceedings of the 17th ACM conference on Information and knowledge management. pp. 509–518. ACM, NY, USA (2008)
16. Ng, V.: Learning noun phrase anaphoricity to improve coreference resolution: issues in representation and optimization. In: ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. pp. 151–158 (2004)
17. Ng, V.: Semantic class induction and coreference resolution. In: Proceedings of the ACL. vol. 45, pp. 536–543 (2007)
18. Ng, V.: Supervised noun phrase coreference research: The first fifteen years. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 1396–1411. Association for Computational Linguistics, Uppsala, Sweden (July 2010), http://www.aclweb.org/anthology/P10-1142
19. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. pp. 104–111 (2002)
20. Ponzetto, S.P., Strube, M.: Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In: Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. pp. 192–199 (2006)
21. Poon, H., Domingos, P.: Joint inference in information extraction. In: AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence. pp. 913–918 (2007)
22. Poon, H., Domingos, P.: Joint unsupervised coreference resolution with Markov Logic. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. pp. 650–659 (2008)
23. Riedel, S., Meza-Ruiz, I.: Collective semantic role labelling with markov logic. In: Proceedings of the Twelfth Conference on Computational Natural Language Learning. pp. 193–197 (2008)
24. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
25. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. Computational Linguistic 27(4), 521–544 (2001)
26. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: WWW '07: Proceedings of the 16th international conference on World Wide Web. pp. 697–706. ACM Press (2007)
27. Versley, Y., Ponzetto, S.P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., Moschitti, A.: Bart: a modular toolkit for coreference resolution. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies. pp. 9–12 (2008)
28. Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: MUC6 '95: Proceedings of the 6th conference on Message understanding. pp. 45–52 (1995)