# Using Background Knowledge to Support Coreference Resolution

**Volha Bryl** and **Claudio Giuliano** and **Luciano Serafini** and **Kateryna Tymoshenko** [1]

**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 direction 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 constitute 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 semantics 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 ACE 2005 dataset, how the injection of knowledge is correlated to the improvement of the performance of our approach on this tasks.

## 1 Introduction

The task of coreference resolution consists in identifying noun phrases (or mentions) that refer to the same real-world entity. For example, 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 [21].

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. [5, 11]).

The second research line investigates the usage of semantic knowledge sources to augment the feature space. Here the majority of the approaches exploit WordNet[2] and, more recently, Wikipedia[3] or corpora annotated with semantic classes (e.g. [13, 15]) to define semantic features, e.g. the semantic relations and the semantic similarity between two mentions.

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 semantics. However, the extension of state-of-the-art coreference methods with these resources is not a trivial task due to the following reasons:

- the *heterogeneity* and the *ambiguity* of the schemes adopted by the different resources of the Semantic Web. This means, for instance, that the same relation can be encoded by different URIs, and that URIs are used by different resources for denoting different relations.
- 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 coreference, while for other entities there is no knowledge at all.
- 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 methodology for coreference resolution that exploits background knowledge available in the web, by proposing three practical solutions of the beforementioned problems:

- To tackle the first problem, we propose a method to map terms in text to URIs through DBpedia [2] mediation. Since most of the resources available in the Semantic Web are linked to DBpedia, we can use it as a *semantic mediator*. So we propose to link text with DBpedia entries and then to exploit the linking between DBpedia and the other resources to access the knowledge encoded in them. DBpedia 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 DBpedia one of the central interlinking hubs of the emerging Web of Data.
- To tackle the issue of selecting the subset of knowledge relevant for coreference, we propose to include only the knowledge that

[1] Fondazione Bruno Kessler, via Sommarive 18, 38123 Trento, Italy

[2] `http://wordnet.princeton.edu/`

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

relates two or more entities of the same document, and knowledge related to some syntactic feature. For the first type of knowledge, for instance, we consider the class-membership relation (e.g. we select the knowledge $President(Barack\_Obama)$, when the text contains "president" and "Barack Obama"), aliases relation (e.g. we select $USA = United\_States$ when the text contains both "the Unites States" and "USA") and so on. As far as knowledge connected with syntactic features we select, for instance, axioms about gender, like $wife(x) \rightarrow female(x)$.

- The problem of integration of statistical (feature-based) information together with background knowledge expressed in RDF/OWL formalism, has been tackled by using an inference engine that support uncertain reasoning. We select the Alchemy tool [1] since it allows 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, which are reported in Section 5. The results show that our method performs in the order of the state-of-the-art coreference algorithms, and, what is more important, that there is a correlation between the presence of the background knowledge and the improvement of performance. This allows us to draw two types of conclusions. First, using background knowledge provides a tangible advantage for coreference resolution, and second, by using the methodology presented in this paper, more improvement could be obtained by simply making available new background knowledge to the system.

## 2 Related work

Soon et al. [21] propose a machine learning framework for coreference resolution, which has become a basis for many later approaches [13, 15, 25]. Soon et al. propose a set of twelve features: lexical, grammatical, positional and semantic. The latter includes the semantic class agreement and alias features. Semantic classes of mentions are obtained from WordNet, alias feature is calculated only for pairs of named entities. They achieve precision of 67.6%, recall of 58.6% and F-measure of 62.60% on the MUC-6 data set, and precision, recall and F-measure of 65.5%, 56.1% and 60.64% on MUC-7, correspondingly.

In the work by Ng and Cardie [13] all the feature subsets from [21] are expanded to 56 features, including the new semantic features obtained from WordNet based on ancestor-descendency relationship and graph distance between mentions. However, experiments with the full feature set show the decrease in the precision on the common nouns to 40.1%. To improve the performance, a number of features were discared, among which there are the semantic ones. With reduced set of features precision and recall are 74.9% and 64.1% on the MUC-6 dataset, and 70.8% and 57.4% on MUC-7.

According to Ng [12], approaches like [21] and [13] assign to a common noun the most frequent sense from WordNet, which may be the reason why in [21] the semantic class agreement feature has zero F-measure. To assign semantic classes to mentions, Ng trains a classifier on the BBN entity corpus. They propose to use the obtained semantic classes both as features and constraints in eight different ways, thus improving precision of the common noun resolution by 2-6% over [21].

Poesio et al. [14] resolve the coreferences that cannot be resolved just by the string matches. They use machine learning techniques to find the best combination of local focus features and lexical distance features, which were calculated using Google API and WordNet 1.7.1. Google features were based on the frequency of predefined patterns, which indicate the coreference. Features from WordNet were based on its hypernym structure. They obtained F-measure of 79.6% using the WordNet features and 77.7% using the Google features.

Many approaches use Wikipedia as a source of semantic information. Yang and Su [25] exploit Wikipedia to extract the patterns, which indicate the semantic relatedness. They add pattern based features to the feature set of [21], thus improving the recall up to 4.3% and F-measure up to 2.1% on the ACE 2004 data set.

Ponzetto and Strube [15] expand the semantic feature subset of [21] by adding two semantic similarity features obtained from WordNet taxonomy and six features obtained from the Wikipedia article texts and category structure. They improve F-measure by 3.4% over the baseline [21] on the ACE 2003 (BNEWS/NWIRE) dataset. To find the correct Wikipedia articles for the mentions the authors query Wikipedia for pages titled as the head lemma. If the disambiguation page is hit, they use an heuristic algorithm. However, the Wikipedia search engine, when queried for a term, very often returns an article about its most frequent sense.

Haghighi and Klein [9] propose a modular approach. In one of the modules they check mention pairs for compatibility. For this purpose they create corpora from 25k articles of the English Wikipedia and 1.8 million sentences of a newswire. It helps them to improve the pairwise F1 from 55.5% to 58% on the ACE2004-ROTH-DEV corpus over other non-semantic modules of their system.

Poesio et al. [23] propose BART, a modular toolkit for coreference resolution. In the feature extraction module the semantic features are the features from [21] and [15]. They reach 65.8% F-measure on MUC-6 and 62.9% F-measure om MUC-7.

Other possible source of semantic information is a knowledge base system Wikitology 2.0. Finin et al. [7] constructed it on the basis of information from Wikipedia and structured knowledge from DBpedia and FreeBase. They use Wikitology 2.0 to solve the ACE task of cross-document coreference resolution. Finin et al. extract intra-document entities using the BBN Serif System. They transform entities into the so-called entity documents EDOCS, which contain various information about the entity's mentions. EDOCS are mapped to Wikitology 2.0. For a given entity the knowledge base returns the vector of matches against Wikipedia article entries and the vector of matches against Wikipedia categories. Finin et al. define twelve Wikitology features based on similarity measures of article/category vectors. Evaluation was performed on the ACE 2008 dataset.

## 3 Background Knowledge Acquisition

This section describes how we train and evaluate a system for acquiring background knowledge from resources linked to DBpedia.

### 3.1 Linking to DBpedia

In order to acquire background knowledge from the Semantic Web, we need to link each mention in a given text to a DBpedia entry and then to exploit the existing links among DBpedia and the other Web resources (e.g., YAGO [22], an ontology extracted from Wikipedia and unified with WordNet) to access the knowledge encoded in them. The linking problem is casted as a word sense disambiguation (WSD) exercise, in which each mention in text (excluding pronouns) has 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 [3]. Notice that linking to Wikipedia entails linking to its structured twin DBpedia, consequently from now on we use the terms Wikipedia page and DBpedia entry interchangeably. The proposed approach is summarized as follows.

#### *3.1.1 Training Set*

To create the training set, for each mention $m$, we collect from the English Wikipedia dump all contexts where $m$ is an anchor of an internal link.[4] The set of target articles represents the senses of $m$ in DBpedia and the contexts are used as labeled training examples. For example, the proper noun *Bush* is a link anchor in $17,067$ different contexts that point to 20 different DBpedia entries, `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. For example, 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 DBpedia entry `Bush_(band)`, its label.

#### *3.1.2 Learning Algorithm*

To disambiguate mentions in text, we implemented a kernel-based approach like in [8]. Different kernel functions are employed to integrate syntactic, semantic, and pragmatic knowledge sources typically used in the WSD literature. The strategy adopted by kernel methods consists of 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 combination of gap-weighted subsequences, bag-of-words, and latent semantic kernels [20].

**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.

**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.

**Latent semantic kernel.** To overcome the drawback of the bag-of-words, we incorporate sematic 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.

#### *3.1.3 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.[5] To classify each mention in DBpedia entries, we used a LIBSVM package.[6] No parameter optimization was performed.

### 3.2 Evaluation

For evaluation, we use a subset of the English ACE 2005 training set[7], which comprises 9 documents with 353 proper nouns. We restricted the evaluation to proper nouns as YAGO, our source of background knowledge, has a limited coverage of common nouns. We carried out the evaluation by manually checking the DBpedia link assigned by the WSD system. The evaluation showed that the WSD system achieved precision, recall, and $F_1$ of 85%, 91%, and 88%, respectively. The baseline system based on the most frequent heuristic achieved precision, recall, and $F_1$ of 82%, 88%, and 85% respectively. In addition, we conducted an error analysis. We discovered that 37% of the errors are due to missing DBpedia entries, 31% to lack of training data, and 32% to classification errors.

## 4 Coreference Resolution with Background Knowledge

In this section we explain how we have implemented the framework to test the main hypothesis of the paper: whether the use of background knowledge obtained from the structured Semantic Web resources improves the performance of NLP tasks, namely, the coreference resolution task. The key choice here is the selection of an inference tool to be used for a task. When the tool is selected, its inputs need to be constructed, that is, we need to define a model for solving the task and find out how the text corpus (enriched with background knowledge as described in the previous section) is to be processed.

### 4.1 Tool selection: Alchemy

A recently introduced family of approaches to the tasks 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 [17, 10, 4] based on the formalism called Markov logic [6], 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 Markov logic framework is competitive in solving NLP tasks (see, for instance, [16, 19, 18], and [1] for more references). Another advantage of the weighted first-order representation is that the model can be easily

[4] A context corresponds to a line of text in the Wikipedia dump and it is represented as a paragraph in a Wikipedia article.

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

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

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

extended with extra (background) 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.

A key concept in Markov logic is the one of Markov logic network, which is a set of pairs $(F_i, w_i)$, where $F_i$ is a first-order formula and $w_i$ is a real number. Together with a set of constants, it defines a Markov network, which contains a node for each possible grounded predicate, with the value of a node equal to 1 if the predicate is true and 0 otherwise. There is an edge between two nodes in the network if the corresponding grounded predicates appear together at least in one grounding of at least one formulae $F_i$. A clique in such a graph corresponds to a grounded formula. A feature $f_i(x)$ is associated to a clique, and has the value is 1 if the corresponding grounded formula holds and 0 otherwise. The Markov logic network defines the joint probability distribution over possible worlds $x$ (where $x$ is a set of values of all the grounded predicates in the network) as follows:

$$P(X = x) = \frac{1}{Z} exp \left( \sum_{j=1}^{F} w_j f_j(x) \right), f_j(x) \in \{0, 1\}$$

where $F$ is the number of formulae in the MLN and $n_i(x)$ is the number of true groundings of $F_i$ in $x$. To perform inference on Markov logic models, Alchemy combines weighted satisfiability (SAT) solvers and Markov chain Monte Carlo inference technique for graphical models [6].

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 named entities in the text, and the properties they might have are gender, number, distance, semantic class, etc. In the two following subsections we discuss in details how these two parts of input are constructed. As an output, the Alchemy inference module produces a list of all possible coreference pairs with associated probabilities. The post-processing of the output is discussed in Section 4.4.

## 4.2 Markov logic model

In defining a model for coreference resolution, we were inspired with Soon et. al. baseline [21], 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 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 named entities 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). Finally, 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[8].

String matching or overlap 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:

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

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

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

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

We also define hard constraints, that is, crisp first-order formulae that should hold in any given world, for instance:

$$20\ singular(x) \wedge plural(y) \rightarrow \neg corefer(x, y)$$
$$\neg corefer(x, x).$$
$$corefer(x, y) \wedge \rightarrow corefer(y, x).$$

Fullstop after the formula refers to an infinite weight, which, in turn, means that the formula holds with the probability equal to 1.

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.

In addition to the syntactic predicates and rules described above, we introduce a set of predicates and rules that deal with background knowledge extracted from a structured Semantic Web knowledge source, YAGO ontology [22] in our case. In this paper, we used just two pairwise semantic properties of mentions: semantic type match and a sort of alias feature derived from YAGO *means* relations (e.g. "United States" may be also referred to as "US", "America", etc.). We define type match relation for proper name/common noun pairs only (e.g. one of the types of the proper name "Obama" matches with a common noun "president"), and introduce also the *unique match* predicate, which describes the situation in which the proper name (the first argument of the predicate) is the only one in the whole document to have a given type. For instance, if a document talks about "Obama" and "Clinton" unique type match with the common noun "president" does not hold for neither of the proper names. The Markov logic model is extended with the a number of rules relating these semantic predicates with coreference property. The arguments of a semantic predicate should be of the same named entity type (person, location, facility, etc.). Non-unique type match property is combined with the *follow* distance relation.

## 4.3 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, coreference resolution task is performed on the 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.

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

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[9], etc.). This allowed us to extract the syntactic properties of the mentions such as number, gender (proper names in the corpus were annotated based on male/female name lists), parwise distance and pronoun and proper name property. For gender, we also defined two lists of tokens (which included “man”,“girl”, “wife”, “Mr.”, etc.).

We worked on the gold standard annotation for named entities, and considered five named entity types: PERson, LOCation, FACility, GeoPoliticalEntity and ORGanization.

As already mentioned above, for extracting semantic properties of the named entities, as a source of background knowledge, we use YAGO [22], an ontology extracted from Wikipedia and unified with WordNet. YAGO ontology contains 1 million entities and 5 million facts. To extract knowledge from YAGO for a given mention, we used the DBpedia link assigned to this mention. The information we extracted from YAGO in this first experiment concerns the *type* and *means* facts about YAGO concepts. Namely, for every *(proper name, noun)* pair of the named entities of the same type we compare the proper name YAGO types with the noun token. In case of the match, the YAGO type match property for a pair is set to true. Differently, means property is extracted for all relevant pairs of named entities.

### 4.4 Alchemy inference and post-processing

We perform Alchemy inference separately for each named entity type (PER, LOC, FAC, GPE, ORG), and then combine the results. 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 $N$x$N$ possible pairs of mentions, where $N$ is the number of mentions:

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

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 [24] can be calculated.

The resulting output of the whole approach includes a list of coreference chains for each document in the corpus, and the measures of the efficiency of the approach, namely, the concrete values of recall, precision and their harmonic mean (F1). We discuss the evaluation of the efficiency in next section.

## 5 Evaluation

Table 1 presents MUC scores of the experiments without and with the use of background knowledge extracted from YAGO for the whole ACE data set (598 documents) for all five types of named entities (ALL), for geopolitical entities (GPE) and persons (PER), accordingly. The improvement in $F1$ for the whole corpus is around 2%. Notice that the recall is improved by 5%, whereas precision goes down by almost 2%. For GPE named entities the improvement is around 3%, while for persons it is just around 1.5%. Lower improvement was achieved for the other three NE types (locations, organizations and facilities), so we do not report these results here. We consider such an improvement to be promising, given that only two YAGO properties were exploited, *type* and *means*, and the possibility to extract and use background knowledge relevant for common nouns was not explored.

| NE type | YAGO | R | P | F1 |
|---|---|---|---|---|
| ALL | no | 0.7272 | 0.8230 | 0.7722 |
| ALL | yes | **0.7778** | 0.8053 | **0.7913** |
| GPE | no | 0.7499 | 0.9404 | 0.8344 |
| GPE | yes | **0.8588** | 0.8631 | **0.8610** |
| PER | no | 0.6989 | 0.7447 | 0.7211 |
| PER | yes | **0.7205** | 0.7495 | **0.7347** |

**Table 1.** MUC scores for all, GPE and PER NE types

Moreover, we have evaluated the dependency between the coverage of the extracted background knowledge on the corpus and the improvement in coreference resolution performance. Tables 2 and 3 report the results for GPE named entity type and *means* YAGO relation, and PER named entity type and *type* YAGO relation, accordingly. Note that the coverage, which is calculated here as number of extracted alias/type matches divided by the total number of pairs in a document, is relatively low. This is related to the observation we made about the potential ways of extending the coverage.

In the tables, **#docs** stands for the total number of documents having a coverage in a given range, and **R-d** (**F1-d**) stands for the difference between recall (F1) with and without background knowledge extracted from YAGO. Recall, precision and F-measure for the cases of absence/presence of the background knowledge are reported in pairs in **R**, **P** and **F1** columns, accordingly. We observe that with the grow of the coverage both recall and F1 generally increase, which supports our hypotheses of the use of background knowledge (extracted from the structured Semantic Web resources) being a promising direction for coreference resolution tasks and, hopefully, for other NLP tasks.

| %cov | #docs | R | R-d | P | F1 | F1-d |
|---|---|---|---|---|---|---|
| 0–2 | 56 | 0.7462<br>0.8491 | **0.1030** | 0.9281<br>0.9109 | 0.8272<br>0.8789 | **0.0517** |
| 2–4 | 52 | 0.7566<br>0.8715 | **0.1149** | 0.9427<br>0.9318 | 0.8395<br>0.9006 | **0.0611** |
| 4–6 | 39 | 0.7583<br>0.8833 | **0.1250** | 0.9550<br>0.9345 | 0.8454<br>0.9082 | **0.0628** |
| 6–10 | 36 | 0.7855<br>0.8934 | **0.1079** | 0.9583<br>0.9404 | 0.8633<br>0.9163 | **0.0530** |
| 10–14 | 16 | 0.7407<br>0.9383 | **0.1975** | 1.000<br>0.8941 | 0.8511<br>0.9157 | **0.0646** |
| 14–28 | 11 | 0.6974<br>0.9079 | **0.2105** | 0.9815<br>0.9718 | 0.8154<br>0.9388 | **0.1234** |

**Table 2.** GPE, correlation between *means* coverage and R/F1 improvement

## 6 Conclusion and future work

In this paper we have defined a methodology for combining semantic information available in the web under the form of logical theories, with statistical methods for natural language processing tasks. The first problem we solved in order to empower an NLP task with the knowledge from publicly available large scale knowledge sources, concerns the mapping of terms in the text to concepts in DBpedia, and then, to other knowledge resources linked to DBpedia, e.g. YAGO ontology. 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, and

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

| %cov | #docs | R | R-d | P | F1 | F1-d |
|---|---|---|---|---|---|---|
| 0–1 | 121 | 0.6877<br>0.7003 | **0.0126** | 0.7245<br>0.7278 | 0.7056<br>0.7138 | **0.0081** |
| 1–2 | 67 | 0.7058<br>0.7320 | **0.0263** | 0.7294<br>0.7366 | 0.7174<br>0.7343 | **0.0169** |
| 2–4 | 60 | 0.6993<br>0.7453 | **0.0460** | 0.7875<br>0.7980 | 0.7408<br>0.7707 | **0.0299** |
| 4–7 | 33 | 0.7327<br>0.7788 | **0.0461** | 0.7852<br>0.7953 | 0.7580<br>0.7870 | **0.0289** |
| 7–21 | 18 | 0.8079<br>0.8821 | **0.0742** | 0.9024<br>0.9058 | 0.8525<br>0.8938 | **0.0413** |

**Table 3.** PER, correlation between *type* coverage and R/F1 improvement

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. We have evaluated the results on the ACE 2005 data set to show the correlation between introducing the new semantic knowledge and the improvement of the performance.

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.

Future work directions include, in the first place, further exploiting YAGO ontology to extract more properties and rules to support coreference resolution. 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. Exploiting other knowledge sources (e.g. Cyc[10] or Freebase[11]) and testing the proposed reference methodology on the other NLP task, like semantic relation extraction, are the other challenging future work directions.

## 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] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In *ISWC/ASWC*, pages 722–735, 2007.

[3] Silviu Cucerzan. 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)*, pages 708–716, 2007.

[4] Aron Culotta, Michael L. Wick, and Andrew McCallum. First-order probabilistic models for coreference resolution. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 81–88, 2007.

[5] Pascal Denis and Jason Baldridge. 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*, pages 236–243, 2007.

[6] Pedro Domingos, Stanley Kok, Daniel Lowd, Hoifung Poon, Matthew Richardson, and Parag Singla. Markov logic. In *Probabilistic Inductive Logic Programming*, volume 4911 of *Lecture Notes in Computer Science*, pages 92–117. Springer, 2008.

[7] Tim Finin, Zareen Syed, James Mayfield, Paul McNamee, and Christine Piatko. Using wikitology for cross-document entity coreference resolution. In *Proceedings of the AAAI Spring Symposium on Learning by Reading and Learning to Read*, 2009.

[8] Claudio Giuliano, Alfio Massimiliano Gliozzo, and Carlo Strapparava. Kernel methods for minimally supervised wsd. *Computational Linguistics*, 35(4):513–528, 2009.

[9] Aria Haghighi and Dan Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1152–1161, 2009.

[10] Shujian Huang, Yabing Zhang, Junsheng Zhou, and Jiajun Chen. Coreference resolution using Markov Logic Networks. In *Proceedings of CICLing 2009*, 2009.

[11] Vincent Ng. 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*, page 151, 2004.

[12] Vincent Ng. Semantic class induction and coreference resolution. In *ACL*. The Association for Computer Linguistics, 2007.

[13] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 104–111, 2002.

[14] Massimo Poesio, Rahul Mehta, Axel Maroudas, and Janet Hitzeman. Learning to resolve bridging references. In *ACL*, pages 143–150, 2004.

[15] Simone Paolo Ponzetto and Michael Strube. 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*, pages 192–199, 2006.

[16] Hoifung Poon and Pedro Domingos. Joint inference in information extraction. In *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pages 913–918, 2007.

[17] Hoifung Poon and Pedro Domingos. Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 650–659, 2008.

[18] Sebastian Riedel and Ivan Meza-Ruiz. Collective semantic role labelling with markov logic. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 193–197, 2008.

[19] Stefan Schoenmackers, Oren Etzioni, and Daniel S. Weld. Scaling textual inference to the web. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–88, 2008.

[20] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[21] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistic*, 27(4):521–544, 2001.

[22] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM Press, 2007.

[23] Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. Bart: a modular toolkit for coreference resolution. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 9–12, 2008.

[24] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *MUC6 '95: Proceedings of the 6th conference on Message understanding*, pages 45–52, 1995.

[25] Xiaofeng Yang and Jian Su. Coreference resolution using semantic relatedness information from automatically discovered patterns. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 528–535, June 2007.

[10] `http://www.cyc.co`

[11] `http://www.freebase.com/`