SANAPHOR: Ontology-Based Coreference Resolution

Roman Prokofyev[1], Alberto Tonon[1], Michael Luggen[1], Loic Vouilloz[2],
Djellel Eddine Difallah[1], and Philippe Cudré-Mauroux[1]

[1] eXascale Infolab, University of Fribourg—Switzerland
[2] Linguistics Department, University of Fribourg—Switzerland
{firstname.lastname}@unifr.ch

Abstract. We tackle the problem of resolving coreferences in textual content by leveraging Semantic Web techniques. Specifically, we focus on noun phrases that coreference identifiable entities that appear in the text; the challenge in this context is to improve the coreference resolution by leveraging potential semantic annotations that can be added to the identified mentions. Our system, SANAPHOR, first applies state-of-the-art techniques to extract entities, noun phrases, and candidate coreferences. Then, we propose an approach to type noun phrases using an inverted index built on top of a Knowledge Graph (e.g., DBpedia). Finally, we use the semantic relatedness of the introduced types to improve the state-of-the-art techniques by splitting and merging coreference clusters. We evaluate SANAPHOR on CoNLL datasets, and show how our techniques consistently improve the state of the art in coreference resolution.

1 Introduction

Natural language understanding is often referred to as an *AI-complete* task, meaning that it belongs to the class of the most difficult problems in Artificial Intelligence, which would require machines to become as intelligent as people prior to being solved. While perfect natural language understanding is still out of reach, recent advances in machine learning, entity linking, and relationship mining are closing the gap between humans and machines when it comes to processing natural language. Semantic technologies have played a key role in those developments, by providing mechanisms to classify, describe, and interrelate entities using machine-processable languages.

Less attention has however been given to the problem of leveraging Semantic Web techniques and knowledge bases to find all expressions referring to the same entity in a text, i.e., *coreference resolution*. While a flurry of previous contributions have proposed techniques to resolve coreferences (see the Related Work section below), the extent to which semantic technologies can be leveraged in this context remains unclear. In this paper, we investigate this question and introduce SANAPHOR, a new system focusing on the last stage of a typical coreference resolution pipeline and improving the quality of the coreference clusters by exploiting semantic entities and fine-grained types to split or merge coreference clusters.

The following piece of text, for example, motivates our approach:
"Laiwu City of Shandong Province has established a cell structure cultivation center ... currently Shangong has established ten agricultural development and model zones similar to that of Laiwu City."
With purely syntactic and grammatical approaches, it is easy to get confused between the name of the province and the name of the city, since they initially appear together. In fact, `Stanford Coref` will put occurrences of both the province and the city into one coreference cluster. Access to external knowledge such as ontologies or knowledge bases is key in this context.

In the following, we add a semantic layer on top of the prominent `Stanford Coref` pipeline[3] to tackle such cases. Throughout our process, we leverage a number of state-of-the-art Semantic Web techniques ranging from entity linking to type ranking. We concentrate on type-based coreferences, excluding part-of-speeches that do not bare self-contained semantics (e.g. determiners, pronouns etc).

In summary, the contributions of this work are:

- A new system that adds a semantic layer to the state-of-the-art `Stanford Coref` pipeline.
- A novel NLP technique that leverages the semantic web to better resolve coreferences.
- An empirical evaluation of our system on standard datasets showing that our techniques consistently improve on the state-of-the-art approach by tackling those cases where semantic annotations can be beneficial.

The rest of this paper is structured as follows: in the rest of this section we define the concepts of coreference and anaphora by presenting several examples; in Section 2 we discuss related work in Semantic Web technologies and on coreference resolution systems; Section 3 describes the architecture of the system we propose; finally, Sections 4 and 5 describe the experimental evaluation of `SANAPHOR` and conclude the paper.

1.1 Preliminaries

We start below by introducing the terminology used throughout the rest of this paper. Some of the linguistic units appearing in textual contents have the function of representing physical or conceptual objects. Linguists often call such units *referring expressions*, while the objects are called *referents* and the relations that unite a referring expression and its referent are called *references*. In the following example: *So Jesus said again, "I assure you, I am the gate for the sheep. All those who came before me were thieves and robbers. [...] I have other sheep too. They are not in this flock here."* the referring expressions are:

[3] `http://nlp.stanford.edu/projects/coref.shtml`

- Noun Phrases (NPs) and pronouns referring to people (e.g. *Jesus* ; *all those who came before me*), things (*the gate*), classes (*sheep*; *they*) or that designate interlocutors (*I*; *you*)
- clauses, that names facts (*I am the gate for the sheep*; *I have other sheep too*; *they are not in this flock here*)
- the adverb *here* that designates a location.

In order to satisfy cohesion [14], the same object is often recalled throughout the text repeatedly so that it can be enriched with new attributes.

In this context, linguists often distinguish *coreference* from *anaphora*. The difference between the two concepts is subtle and is explained in the following. We have a *coreference* every time two (possibly different) referring expressions denote *the same referent*, that is, the same entity. For example, in the sentence *Abraham Lincoln, the first president of the USA, died in 1865.*, "Abraham Lincoln" and "the first president of the USA" refer to the same entity, thus, they co-refer. We have an *anaphora* every time the reference of an expression $E2$, called *anaphoric expression*, is function of a previous expression $E1$, called *antecedent*, so that one needs $E1$ to interpret $E2$. For example, in the sentence *I like dragons! Those animals are really cute!* "those animals" is an anaphoric expression and the reader needs to know that it refers to "dragons" (the antecedent) in order to understand the sentence. Finally, the two concepts can be combined:

- The sentence *You have a cat? I don't like them.* is a case of anaphora without coreference since the pronoun *them* needs the antecedent *a cat* to be interpreted (it is the anaphoric), but the two references do not designate the same object (*a cat* = an individual / *them* = the entire species).
- The sentence about Abraham Lincoln we presented before is an example of coreference without anaphora, since if we remove "Abraham Lincoln" one can still understand the sentence.
- The sentence *The dragon is coming. It is going to burn the city!* is an example of anaphora and coreference since one needs an antecedent to resolve "It", and both "It" and "the dragon" refer to the same entity.

In this paper we show how entity types can be used in order to resolve the two last cases.

2 Related Work

2.1 Named Entity Recognition

Named entity recognition (NER) refers to the task of correctly identifying words or phrases in textual documents that represent entities such as people, organizations, locations, etc. During the last decades, NER has been widely studied and the best NER approaches nowadays produce near-human recognition accuracy for generic domains such as news articles. Several prominent NER systems employ supervised learning methods based on maximum entropy [4] and conditional random fields [8], or fuse the results of other systems using a supervised classifier [33].

2.2 Entity Linking

Entity linking is the task of associating a textual mention of an entity to its corresponding entry in a knowledge base. It can be divided into three subtasks: mention detection, link generation, and disambiguation [21]. One of the main issues that needs to be tackled when doing entity linking is the ambiguity of the textual representation of the entity given as input. For example, the mention "Michael Jordan" can be linked to both Michael Jordan the basketball player and Michael Jordan the well-known machine learning professor. Much work has been done on entity linking. Recently, Houlsby and Ciaramita dealt with ambiguities by using a variant of LDA in which each topic is a Wikipedia article (that is, an entity) [17]. Cheng and Roth used Integer Linear Programming to combine relational analysis of entities in the text, features extracted from external sources and statistics on the text [6].

In the context of this paper, both NER and Entity Linking are prerequisites for coreference resolution as we take advantage of external knowledge to improve the resolution of coreferences and hence must first identify and link as many entity mentions as possible to their counterparts in the knowledge base. Since, however, those two tasks are not the focus of this work, we decided to use in this paper the TRank pipeline because of its simplicity and its good performance in practice on our dataset (see Section 4).

2.3 Entity Types

Knowing the types of a certain entity is valuable information that can be used in a variety of tasks. Much work has been done on extracting entity types both from text and from semi-structured data. In this context, Gangemi *et al.* [9] exploit the textual description of Wikipedia entities to extract entity types, Nakashole *et al.* [24] designed a probabilistic model to extract the types out of knowledge base entities, and Paulheim and Bizer [28] worked on adding missing type statements by exploiting statistical distributions of types as subjects and objects of properties. Much effort has been put also on ranking entity types in several contexts. TRank [38] is a system for ranking entity types given the textual context in which they appear. Tylenda *et al.* [39] select the most relevant types to summarize entities. In this paper we leverage entity types as evidences for deciding if, given a piece of text, different entity mentions refer to the same entity or not.

2.4 Coreference and Anaphora

According to Ng [25], practically all coreference and anaphora resolution systems are instantiations of a seven-step generic algorithm[4]:

[4] Note that steps 3, 5 and 6 can be absent in a coreference or anaphora resolution algorithm. Moreover, existing algorithms differ in the way these seven steps are implemented

1. Identification of referring expressions: This first step is mostly to identify all of the pronouns and noun phrases in the text. Clauses and adverbs can also be spotted.
2. Characterization of referring expressions: This second step consists of determining and computing the information regarding referring expressions that might be relevant to its linking to another expression in the text. Most approaches rely on some preprocessing modules (e.g. part-of-speech tagging, parsing, named entity recognizer,. . .) to perform this step ; however, they differ in the level of sophistication of the extracted information, ranging from knowledge-rich to knowledge-poor (see below).
3. Anaphoricity determination: Involves distinguishing anaphoric expressions, that should have an antecedent, from non-anaphoric expressions, that should not. Thus, this step is always performed as part of anaphora resolution, but not always for coreference resolution (see 1.1).
4. Generation of antecedent candidates: This fourth step identifies a set of potential antecedents, named *candidates*, that linearly precedes the anaphoric expression in the text.
5. Filtering: This step involves removing from the set some unlikely candidates based on ensemble of hard constraints, for example morphologic, syntactic and semantic constraints.
6. Scoring/Ranking: The aim of this step, that is optional, is to rank remaining candidates according to an ensemble of soft constraints, also called *preferences*, that often depend on psycholinguistic and discourse principles (especially *focus* [34], *centering* [12] or *accessibility* [1]).
7. Searching/Clustering: Finally, the goal of this last step is to select an antecedent for a given anaphoric expression from the set of candidates returned by the fifth and/or the sixth steps. If step 6 has been performed, then *searching* becomes the task of selecting the highest-ranking element in the candidate list; otherwise, the "best" expression is selected as the antecedent in accordance with criteria specified by the resolution algorithm. In the case of coreference resolution, this process corresponds to applying a single-link clustering algorithm to each anaphoric expression to cluster the referring expressions in the document and generate a partition.

Although this generic algorithm characterizes most of the resolution pipelines, research on coreference and anaphora resolution in computational linguistics has been proceeding in many different directions for the last 30 years. Nevertheless, it is possible to identify important trends [25,7,27]. In the context of this paper, two trends are of particular significance and are presented below.

First, coreference and anaphora resolution systems can be classified with respect to the types of knowledge sources they leverage. One typically differentiates *Knowledge-rich* systems from *knowledge-lean systems.* Early anaphora resolution systems [11,35] as well as more recent ones [37,29,26,13,5,40] are knowledge-rich systems that rely on domain informations (such as FrameNet, WordNet, Wikipedia, Yago, etc.), semantic and discourse analysis, and sophisticated inference mechanisms (induction for example). Knowledge-lean systems instead rely

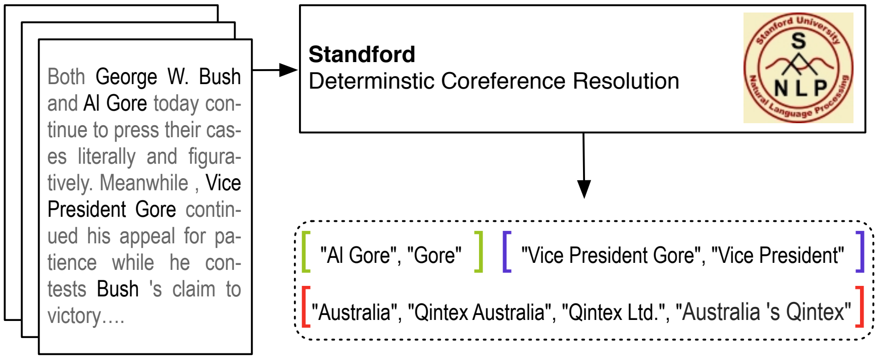
Both George W. Bush and Al Gore today continue to press their cases literally and figuratively. Meanwhile , Vice President Gore continued his appeal for patience while he contests Bush 's claim to victory....
Standford
Determinstic Coreference Resolution
Stanford University
S
N L P
Natural Language Processing
"Al Gore", "Gore"
"Vice President Gore", "Vice President"
"Australia", "Qintex Australia", "Qintex Ltd.", "Australia 's Qintex"

Fig. 1. The `Stanford Coref` system takes plain text as an input and outputs clusters (`[]`) of mentions (`""`) which are potentially coreferenced.

only on morphological and possibly syntactic information [18,3,23,19], and reach high performance without semantic and world knowledge. Our system belongs to the first category, using YAGO and DBpedia.

Early coreference and anaphora resolution systems also differ from more recent ones by the fact that they adopt *knowledge-based approaches*, in which the rulesets used in *filtering* and *scoring/ranking* (see steps 5 and 6 above) are based on a set of hand-coded heuristics that specify whether two referring expressions can or cannot have any coreferential/anaphoric relationship [16,12]. Actually, these approaches are often called *linguistic approaches* as they are based on linguistic theories. In contrast, *corpus-based approaches* acquire knowledge using a learning algorithm and training data, i.e., a corpus annotated with coreference and anaphora information in *filtering* and *scoring/ranking* [10,15,36]. Again, our own system belongs to the first category.

3 System Architecture

In this section, we describe the overall architecture of `SANAPHOR` and provide details on each of its components.

3.1 System Input

Starting from the `Stanford Coref` framework [19] (Figure 1), which covers the steps 1-7 described in Section 2.4, we obtain for each document (e.g., a news article) a set of clusters containing textual mentions. The clusters are non-overlapping and contain potentially coreference mentions. In addition, `Stanford Coref` associates a headword to each mention (especially for long mentions) when possible.

3.2 System Overview

Many potential improvements are conceivable throughout the generic pipeline introduced in Section 2.4. In that context, our efforts first focused on improving coreference resolution using semantic word and phrase similarities based on Word Vectors [22]. However, word vectors did not work well in our experiments. For example, the vector of the word “shepherd” was very close to the vector of “sheep”, which is reasonable, but does not work well for the coreference resolution task, since these two words often appear in one document. Motivated by the results analysis presented above, `SANAPHOR` focuses instead on splitting and merging of candidate clusters (see Step 7 in Section 2.4) using semantic information, as it is (in our opinion) the most susceptible to benefit from a tight integration of semantic technologies.

Figures 2 and 3 give an overview of our system, illustrating the preprocessing steps and the splitting/merging steps respectively. `SANAPHOR` receives as input the clusters of coreferences generated by `Stanford Coref`. Each cluster is a set of mentions extracted from the original text. Each mention comes in the form of a string and, potentially, an associated headword (the most salient word in the mention). The mentions can be either Named Entities, pronouns, or determin-

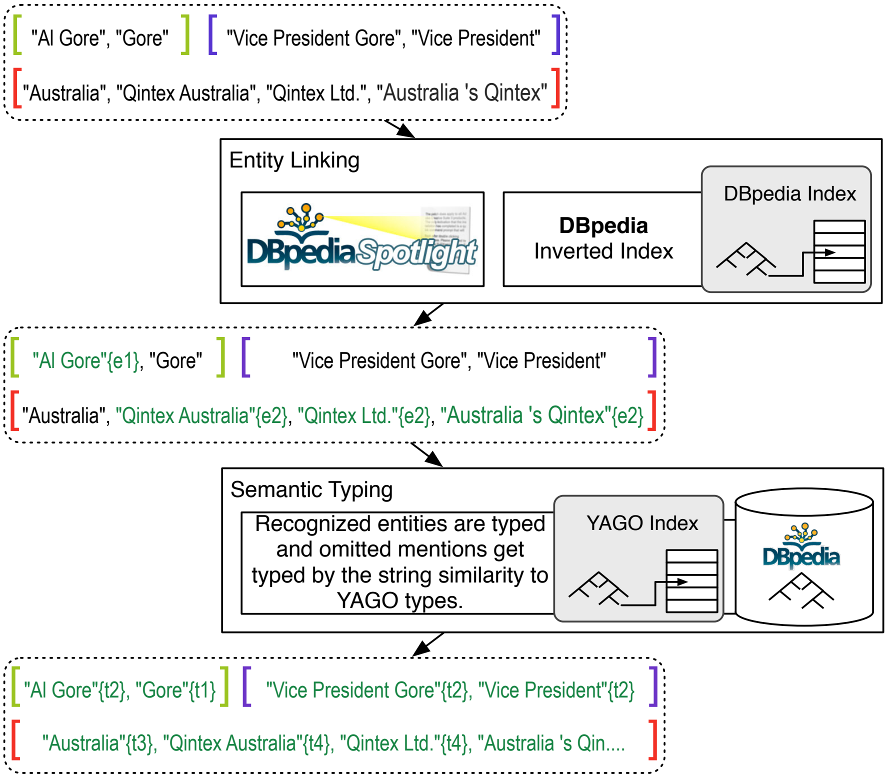
["Al Gore", "Gore"] ["Vice President Gore", "Vice President"]
["Australia", "Qintex Australia", "Qintex Ltd.", "Australia 's Qintex"]
Entity Linking
DBpedia Spotlight
DBpedia Inverted Index
DBpedia Index
["Al Gore"{e1}, "Gore"] ["Vice President Gore", "Vice President"]
["Australia", "Qintex Australia"{e2}, "Qintex Ltd."{e2}, "Australia 's Qintex"{e2}]
Semantic Typing
Recognized entities are typed and omitted mentions get typed by the string similarity to YAGO types.
YAGO Index
DBpedia
["Al Gore"{t2}, "Gore"{t1}] ["Vice President Gore"{t2}, "Vice President"{t2}]
["Australia"{t3}, "Qintex Australia"{t4}, "Qintex Ltd."{t4}, "Australia 's Qin....]

Fig. 2. The pre-processing steps of `SANAPHOR` annotating semantics to the mentions.

ers, as identified and clustered by `Stanford Coref`. Our system then takes those clusters and proceeds in two successive steps I) Preprocessing, where we leverage linked data to represent named entities with their semantic counterparts (either Entities or Types) whenever possible; II) Cluster Optimization, where using annotations obtained from the preprocessing step we derive a strategy for splitting clusters containing unrelated mentions, or, conversely for merging mentions that semantically should belong together.

We describe in more detail the functionalities provided by those components in the following, starting with the semantic annotation pipeline and then moving to cluster management methods.

3.3 Semantic Annotation

Entity Linking The goal of the Entity Linking component is to link entity mentions to DBpedia entries. We exploit an inverted index associating DBpedia labels to entity URIs. In order to generate high-quality links, we decided to

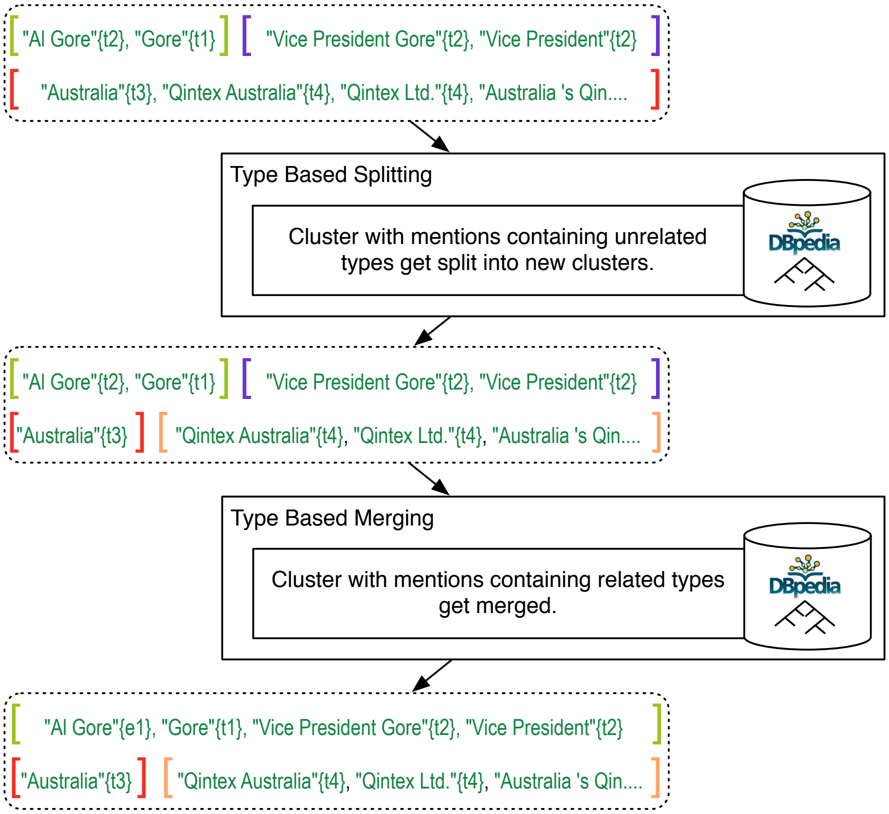
["Al Gore"{t2}, "Gore"{t1}] ["Vice President Gore"{t2}, "Vice President"{t2}]
["Australia"{t3}, "Qintex Australia"{t4}, "Qintex Ltd."{t4}, "Australia 's Qin....]
Type Based Splitting
Cluster with mentions containing unrelated types get split into new clusters.
DBpedia
["Al Gore"{t2}, "Gore"{t1}] ["Vice President Gore"{t2}, "Vice President"{t2}]
["Australia"{t3}] ["Qintex Australia"{t4}, "Qintex Ltd."{t4}, "Australia 's Qin....]
Type Based Merging
Cluster with mentions containing related types get merged.
DBpedia
["Al Gore"{e1}, "Gore"{t1}, "Vice President Gore"{t2}, "Vice President"{t2}]
["Australia"{t3}] ["Qintex Australia"{t4}, "Qintex Ltd."{t4}, "Australia 's Qin....]

Fig. 3. The final type-based splitting and merging of the clusters in `SANAPHOR`.

only link mentions that exactly match DBpedia labels[5]. Entities with multiple aliases are handled by using Wikipedia redirection links and, in order to foster precision, by discarding URIs that link to ambiguous entities (i.e., entities having a `wikiPageDisambiguates` property).

Semantic Typing The next step in our preprocessing pipeline is assigning *Types* to mentions appearing in the text. In this context, we use the YAGO ontology as a target database. We created an inverted index of the types obtained from the YAGO ontology[6] and performed a string matching between every mention and the inverted index. For example, a noun phrase "rock singer" is typed as ⟨Wikicategory_American_Rock_Singers⟩. For the mentions linked in the previous step, we employ the mappings between DBPedia and YAGO ontologies provided by TRank Hierarchy [38] to map DBPedia types to YAGO ones.

We chose to optimize our preprocessing steps for precision rather then recall, since the subsequent steps rely on precise linking to be effective at improving the mention clusters. As a result, we do not annotate labels that refer to multiple entity types.

3.4 Cluster Management

Splitting Coreference Clusters The first task `SANAPHOR` undertakes to optimize the clusters of mentions is to split clusters containing mentions of different types. This step tackles cases where `Stanford Coref` was not able to deal with ambiguity in the text, for example for the following cases: "Aspen"(the Colorado city) and "Aspen"(the tree), which can be wrongly interpreted as referring to the same referent, thus producing a series of incorrect coreferences. Instead, `SANAPHOR` leverages the output of the entity linking process to resolve the ambiguity of the mentions: since during the linking phase the two mentions will probably be associated to different entities, the system can decide to split them into separate clusters.

The result of the semantic annotation phase is a series of sets $\{\mathcal{S}_0, \ldots, \mathcal{S}_n\}$, one per coreference cluster, containing entities $e \in \mathcal{E}$ and/or fined-grained semantic types $t \in \mathcal{T}$ attached to each mention $m \in \mathcal{M}$. The splitting process examines all pairs of mentions $\{m_i, m_j\}$ in a given cluster, and decides whether or not to split the cluster depending on the potential entities $\{e_i, e_j\}$ and types $\{t_i, t_j\}$ attached to the mentions. Formally, we split a cluster whenever, $\forall\{m_i, m_j\} \in S$:

- $\exists\{e_i, e_j\} \mid e_i \neq e_j$ or
- $\exists\{t_i, t_j\} \mid t_i \nleq t_j$ (where $\nleq$ stands for equivalence or subsumption relation w.r.t. the type hierarchy of the ontology), or
- $\exists\{e_i, t_j\} \mid T(e_i) \nleq t_j$ (where $T(e_i)$ stands for the type of e_i according to the ontology).

[5] We have also tried more complex methods that take context into account, such as *DBPedia Spotlight*, but they lead to less precise linkings and worse overall results.

[6] `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/`

Since a coreference cluster might also contain non-annotated mentions, we need a way to properly assign them to the split clusters. In order to do this, we first identify the words that belong exclusively to one of the mentions m_i or m_j. We assign all other mentions to one of the new clusters based on the overlap of their words with the exclusive words of each new cluster.

However, these steps alone do not systematically result in a substantial performance increase due to many possible reductions of the original mention. For example, a text might contain "Aspen Airways" first and then have the word "Aspen" to refer to the airline, which our method might incorrectly link to a city or a tree type. To overcome this problem, we introduce a simple heuristic that ignores entity linkings of the mentions whose words represent a complete subset of any other mention in the same cluster.

Merging Coreference Clusters The second task that we are tackling in the context of cluster management is merging, that is, joining pairs of sets $\{S_i, S_j\}$ that contain similar entities or types. For instance, consider the mention "Hosni Mubarak", the former president of Egypt, which can also be referred to as "President Mubarak" in a news article. In such a case, `Stanford Coref` might assign those two mentions to two different clusters. Thus, starting from entity and type linking as before, we propose to merge clusters, each of which contains at least one mention that refers to the same entity. Formally, two sets $\{S_i, S_j\}$ corresponding to two clusters are merged whenever:

- $\exists \; (e_i \in S_i \wedge e_j \in S_j) \mid e_i \equiv e_j$ or
- $\exists \; (e_i \in S_i \wedge t_j \in S_j) \mid T(e_i) \leq t_j$ and when the condition just above does not apply.

We note that in this step we do not use any heuristic to pre-filter the clusters.

Our system, `SANAPHOR`, is available as an open-source[7] extension to `Stanford Coref`. The pipeline allows to use different entity and type linkers for future experiments.

4 Experimental Evaluation

4.1 Datasets

We evaluate our system on standard datasets from the CoNLL-2012 Shared Task on Coreference Resolution [30] distributed as a part of the OntoNotes 5 dataset[8]. We use only the English part of the dataset which consists of over one million words from newswire, magazine articles, broadcast news, broadcast conversations, web data, telephone conversations and English translation of the New Testament.

[7] `http:///github.com/xi-lab/sanaphor`

[8] `https://catalog.ldc.upenn.edu/LDC2013T19`

The English dataset is split into three: development, training and test sub-collections. The development dataset is intended to be analyzed during the development of the coreference resolution system in order to build intuitions and tune the system. The training dataset is designed to be used in the supervised training phase, while the final results have to be reported on the test dataset. In the following sections, we analyze results and we design our methods based on the development collection and report the final results based on the test collection. Since our system improves on the Stanford Coreference Resolution System, which already includes supervised models, we do not directly use the training sub-collection in our pipeline.

4.2 Metrics

Many metrics have been proposed to evaluate the performance of coreference resolution systems, from early metrics like MUC [41], to the most recent metric proposed—BLANC [32].

As a final evaluation metric, we use the most recently proposed BLANC, which addresses the drawbacks of previously proposed metrics such as MUC, B-cubed [2], or CEAF [20], as it neither ignores singleton mentions nor does it inflate the final score in their presence.

In addition, we use a pairwise metric based on the *Rand Index* [31] to evaluate the performance of the individual parts of our system in isolation.

4.3 Analysis of the Results of Stanford Coreference Resolution System

We start by analyzing the results of the `Stanford Coref` on the development dataset in the context of two possible error classes: 1) mentions that were put into one cluster, but that in fact belong to different clusters, 2) mentions that refer to the same thing, but that were put into different clusters. Additionally, since we focus on noun-phrase mentions, we want to see how many noun-only clusters exist in the dataset in order to estimate the effect of a possible improvement.

Overall, the `Stanford Coref` system creates 5078 coreference clusters, out of which 270 clusters need to be merged and 77 "has-to-be-merged" clusters are noun-only. The total number of clusters that should be split is 118, out of which 52 are noun-only.

As we can observe, the total amount of potential split and merge clusters account for approximately 8% of total data, which can result in a significant performance improvement for coreference resolution (for which even small improvements are considered as important given the maturity of the tools developed over more than 30 years).

In the following, we report results for the different steps in our pipeline on the test dataset.

	0 Links	1 Distinct Link	2 Distinct Links	3 Distinct Links
All Clusters	4175	849	49	5
Noun-Only Clusters	1208	502	33	2

Table 1. Cluster linking distributions for all the clusters and for noun-only clusters

4.4 Preprocessing Results

The main innovation of SANAPHOR is the semantic layer that enhances classic coreference clustering, hence we focus on evaluating clusters that contain at least one entity (or one type) at the output of our preprocessing steps. The overall recall of our approach is therefore bound by the number of clusters that were identified as containing linked entities and/or types.

In total, we linked 2607 mentions out of 9664 noun phrase mentions (i.e., mentions that have nouns as headwords) extracted by Stanford Coref from the **CoNLL dev dataset**. Out of these 9664 mentions, 4384 were recognized by Stanford Coref as entities. Table 1 summarizes the distribution of clusters and the links obtained using our preprocessing step.

For evaluation purposes, we consider only clusters that contain at least one link. Moreover, we make the following distinction of clusters for evaluation purposes:

- **All Linked Clusters.** That is, clusters that contain at least one linked mention, or
- **Noun-Only Linked Clusters.** These are clusters which contain at least one linked mention, headwords, but have no pronouns nor determiners.

We make this distinction in order to evaluate whether considering clusters with pronouns and determiners (which bare little semantic information) affects the overall results.

4.5 Cluster Optimization Results

Now, we turn our attention to the evaluation of the effectiveness of our cluster optimization methods (splitting and merging). The following experiments are performed on the CoNLL test dataset. We compute Precision, Recall and F1 metrics for the clusters on which we operate. Since we are evaluating clusters, we use the pairwise definition of the metrics (see Section 4.2).

We distinguish the results for both the split and merge operations as compared to the ground-truth. For instance, for all the clusters generated by each system, we perform pairwise comparisons of all mentions in the clusters and evaluate whether the two mentions were correctly separated (in case of a split) or put together (in case of a merge).

Table 2 summarizes the results of our evaluation. As can be seen, SANAPHOR outperforms Stanford Coref in both the split and merge tasks for both All and Noun-Only clusters. Moreover, we notice that the absolute

		SANAPHOR			Stanford Coref		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Split	All Clusters	82.56	90.27	**86.25**	71.39	100.00	83.31
	Noun-Only Clusters	78.99	90.38	**84.30**	58.43	100.00	73.76
Merge	All Clusters	94.58	100.00	**97.21**	96.65	55.10	70.18
	Noun-Only Clusters	76.92	100.00	**86.96**	85.00	56.67	68.00

Table 2. Results of the evaluation of the cluster optimization step (split and merge).

increase in F1 score for the split task is greater for the Noun-Only case (+10.54% vs +2.94%). This results from the fact that All Clusters also contain non-noun mentions, such as pronouns, which we don't directly tackle in this work but have to be assigned to one of the splits nevertheless. Our approach in that context is to keep the non-noun mentions with the first noun-mention in the cluster, which seems to be suboptimal for this case.

For the merge task, the difference between All and Noun-Only clusters is much smaller (+27.03% for the All Clusters vs +18.96% for the Noun-Only case). In this case, non-noun words do not have any effect, since we merge clusters and also include all other mentions.

4.6 End-to-End Performance

Finally, and in addition to the previous results that reflect the effectiveness of SANAPHOR on relevant clusters, we evaluate the impact of our approach on the end-to-end coreference resolution pipeline using the CoNLL test collection. In that context, we use the Precision, Recall and F1 scores of the BLANC metric (Section 4.3). Our system consistently outperforms the Stanford Coref baseline in both Precision (60.63% vs 60.61%), Recall (55.16% vs 55.07%) and F1 values (57.11% vs 57.04%). The reason behind the limited improvement on the overall dataset is imputable to the recall we achieve during the linking step (see Section 4.4) and to the limited number of cases in which a split or a merge is required (8% of the total data).

To further elaborate on the significance of our results, we also ran our SANAPHOR pipeline on the data where we annotated all entities with the "gold" (i.e., ground-truth) URLs. This corresponds to the optimal case where the system is able to link all possible entities correctly. The performance of Stanford Coref for such a best-case scenario is 57.17% in terms of F1, which is comparable to the performance of our entity linking method, thus confirming the validity of our approach.

5 Conclusions

In this paper, we tackled the problem of coreference resolution by leveraging semantic information contained in large-scale knowledge bases. Our open-source system, SANAPHOR, focuses on the last stage of a typical coreference resolution pipeline (*searching and clustering*) and improves the quality of the coreference clusters by exploiting semantic entities and fine-grained types to split or merge the clusters. Our empirical evaluation on a standard dataset showed that our techniques consistently improve on the state-of-the-art approach by tackling those cases where semantic annotations can be beneficial.

Our approach can be extended in a number of ways. One of the limitations of SANAPHOR affecting its recall is due to the potential lack of information being available in the knowledge base. In that sense, techniques that take advantage of a series of knowledge bases (e.g., based on federated queries), that identify missing entities in the knowledge base or that dynamically enrich the knowledge base could be developed. Another interesting extension would be to bring more structure to the coreference clusters, for example by introducing semantic links between the candidates in order to foster more elaborate post-processing at the merging step.

6 Acknowledgments

This work was supported by the Swiss National Science Foundation under grant number PP00P2 153023, and by the Haslerstiftung in the context of the Smart World 11005 (Mem0r1es) project.

References

1. Mira Ariel. *Accessing noun-phrase antecedents*. Routledge, 2014.
2. Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer, 1998.
3. Breck Baldwin. Cogniac: high precision coreference with limited knowledge and linguistic resources. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 38–45. Association for Computational Linguistics, 1997.
4. Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. *Sixth Workshop on Very Large Corpora*, chapter Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. 1998.
5. Volha Bryl, Claudio Giuliano, Luciano Serafini, and Kateryna Tymoshenko. Using background knowledge to support coreference resolution. In *ECAI*, volume 10, pages 759–764, 2010.
6. Xiao Cheng and Dan Roth. Relational Inference for Wikification. *Empirical Methods in Natural Language Processing*, pages 1787–1796, 2013.
7. Pradheep Elango. Coreference resolution: A survey. Technical report, University of Wisconsin, Madison, 2005.
8. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

9. Aldo Gangemi, Andrea Giovanni Nuzzolese, Valentina Presutti, Francesco Draicchio, Alberto Musetti, and Paolo Ciancarini. Automatic typing of DBpedia entities. In *The Semantic Web–ISWC 2012*, pages 65–81. Springer, 2012.
10. Niyu Ge, John Hale, and Eugene Charniak. A statistical approach to anaphora resolution. In *Proceedings of the sixth workshop on very large corpora*, volume 71, 1998.
11. Barbara J Grosz et al. The representation and use of focus in a system for understanding dialogs. In *IJCAI*, volume 67, page 76, 1977.
12. Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225, 1995.
13. 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: Volume 3 - Volume 3*, EMNLP '09, pages 1152–1161, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
14. Michael A.K. Halliday and Ruqaiya Hasan. *Cohesion in English.* Longman, London, 1976.
15. Sanda M Harabagiu, Rzvan C Bunescu, and Steven J Maiorano. Text and knowledge mining for coreference resolution. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics, 2001.
16. J Hobbs. Resolving pronoun references. In *Readings in natural language processing*, pages 339–352. Morgan Kaufmann Publishers Inc., 1986.
17. Neil Houlsby and Massimiliano Ciaramita. A scalable Gibbs sampler for probabilistic entity linking. In *Advances in Information Retrieval*, pages 335–346. Springer, 2014.
18. Shalom Lappin and Herbert J Leass. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561, 1994.
19. Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics, 2011.
20. Xiaoqiang Luo. On coreference resolution performance metrics. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32. Association for Computational Linguistics, 2005.
21. Edgar Meij, Krisztian Balog, and Daan Odijk. Entity linking and retrieval. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 1127–1127, New York, NY, USA, 2013. ACM.
22. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.
23. Ruslan Mitkov. Robust pronoun resolution with limited knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, pages 869–875. Association for Computational Linguistics, 1998.
24. Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. Fine-grained semantic typing of emerging entities. In *ACL (1)*, pages 1488–1497, 2013.

25. Vincent Ng. Machine learning for coreference resolution: Recent successes and future challenges. Technical report, Cornell University, 2003.
26. Vincent Ng. Semantic class induction and coreference resolution. In *AcL*, pages 536–543, 2007.
27. Vincent Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1396–1411. Association for Computational Linguistics, 2010.
28. Heiko Paulheim and Christian Bizer. Improving the Quality of Linked Data Using Statistical Distributions. *Int. J. Semantic Web Inf. Syst.*, 10(2):63–86, January 2014.
29. Simone Paolo Ponzetto and Michael Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 192–199. Association for Computational Linguistics, 2006.
30. Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, CoNLL '12, pages 1–40, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.
31. William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.
32. M. Recasens and E. Hovy. Blanc: Implementing the rand index for coreference evaluation. *Nat. Lang. Eng.*, 17(4):485–510, October 2011.
33. Giuseppe Rizzo and Troncy Raphaël. NERD : A Framework for Evaluating Named Entity Recognition Tools in the Web of Data. *Proceedings of the 11th Interational Semantic Web Conference ISWC2011*, pages 1–4, 2011.
34. Candace Sidner. Focusing in the comprehension of definite anaphora. In *Readings in Natural Language Processing*, pages 363–394. Morgan Kaufmann Publishers Inc., 1986.
35. Candace Lee Sidner. Towards a computational theory of definite anaphora comprehension in english discourse. Technical report, DTIC Document, 1979.
36. Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544, 2001.
37. Michael Strube and Simone Paolo Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*, volume 6, pages 1419–1424, 2006.
38. Alberto Tonon, Michele Catasta, Gianluca Demartini, Philippe Cudré-Mauroux, and Karl Aberer. Trank: Ranking entity types using the web of data. In *The Semantic Web – ISWC 2013*, volume 8218 of *Lecture Notes in Computer Science*, pages 640–656. Springer Berlin Heidelberg, 2013.
39. Tomasz Tylenda, Mauro Sozio, and Gerhard Weikum. Einstein: Physicist or vegetarian? summarizing semantic type graphs for knowledge discovery. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 273–276, New York, NY, USA, 2011. ACM.
40. Olga Uryupina, Massimo Poesio, Claudio Giuliano, and Kateryna Tymoshenko. Disambiguation and filtering methods in using web knowledge for coreference resolution. In *FLAIRS Conference*, pages 317–322, 2011.
41. Kees Van Deemter and Rodger Kibble. On coreferring: Coreference in muc and related annotation schemes. *Computational linguistics*, 26(4):629–637, 2000.