MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

Coreference Resolution Sieve Based on Answer Set Programming

KENDA ALAKRAA

MASTER THESIS
Department of Computer Engineering

Thesis Supervisor
Asst Prof. Dr. Peter Schüller

ISTANBUL, 2017

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN PURE AND APPLIED SCIENCES

Coreference Resolution Sieve Based on Answer Set Programming

KENDA ALAKRAA

(524113924)

MASTER THESIS

Department of Computer Engineering

Thesis Supervisor

Asst Prof. Dr. Peter Schüller

ISTANBUL, 2017

Acknowledgments

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this thesis.

I wish to express my special appreciation and my sincere thanks to my advisor Assistant Professor Peter Schüller, who taught me the value of hard work and an education. Although I had little experience, he is the one who believed that I can do master when even sometimes I lost trust in myself. Without him, I may never have gotten to where I am today.

My deepest gratitude goes to my late father Mr. Alakraa Mansour and my mother Mrs. Rahhal Suria for their endless love, prayers, encouragement, and all the support they have provided me over the years.

I owe thanks to a very special person, my husband, for his love, care, support, and understanding that made the completion of thesis possible.

Special and profound thanks to my brothers and sisters, Iyad, Omar, Furat, Ahmad, Najd, and Abdulazeez who offered support and humor over the years,

Many thanks to all my friends Noor, Lina, Ilham, Lama, Dalya, Raneem, Gamze, and Zuhal.

To Şadan Safranbolu abla, I can not find words to thank you "Allah sizden razi olsun ablam".

To those who indirectly contributed in this research. Thank you very much.

This work has been supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under grant 114E430.

Contents

Özet

Çözüm Kümesi Programlama tabanlı Eşgönderge Çözümlenmesi Eleği

Coreference (eşgönderge) çözümlenmesi, cümleleri ve aynı gerçek dünya varlığını ifade ettikleri takdirde metin içindeki edatları bağlama işine verilen isimdir. Bu görevde yüksek başarı elde etmek bilgi ve semantik gerektirir, örneğin; "Obama" ve "başkan" kelimeleri arasında bağlantı kurulmalı ya da içerik bilgisine bağlı olarak, mesela bu bilgi makalenin tarihi olabilir, o takdirde bu iki kelime arasında bağlantı kurulmamalıdır. Modern son teknoloji coreference çözümlenmesi methodları istatistiksel yöntemleri veya kuralları veya her ikisini birden kullanır, ancak nadiren metnin yüzeyinin ötesine geçer. Bu çalışmada Stanford "dcoref" coreference çözümlenmesi mimarisinin, bilginin entegrasyonunu sağlayacak hesaplama mantığı kullanarak uyarlanmasını önerilmektedir. Stanford Sieve deterministik mimarisini uygulamak için temel bir kural çerçevesi tanımlanmaktadır, bu da deterministik parçayı gerçekleştirmenin uygun olduğunu göstermektedir. Yaptığımız iş, deterministik mimariyi nondeterministik hale getirmeye yönelik ilk adımdır. Vizyonumuz, yeni mimarinin, (i) bağlantılar üzerinde kısıtlamalar getirmesi ve (ii) sığ semantik bilgiye dayalı olarak keşfedilemeyen bağlantılar için yeni adaylar üretebilmesi için, yeni mimarinin anlambilimsel bilgiyi bütünleştirmesine olanak tanımasıdır. Çerçevemizle elde ettiğimiz deneysel sonuçlar, hesaplama mantığının istenen esnekliği sağladığını, bununla birlikte çok daha fazla kaynak gerektirdiğini önermektedir. Bu, kuralları daha az sürdürülebilir hale getiren performans kurallarını ayarlayarak önlenebilir ve ileride otomatik kural tabanlı optimizasyon algoritmalarının geliştirilmesinde kullanılması önerilmektedir .

Keywords: Eşgönderge çözümlenmesi, Sieve sistemi, Stanford CoreNLP.

Abstract

Coreference Resolution Sieve Based on Answer Set Programming

Coreference Resolution is the task of connecting phrases and prepositions in a text if they denote the same real world entity. Succeeding with a high score in this task naturally requires knowledge and semantics, for example to link “Obama” with “the president”, or, depending on the context, for example the date of the article, not to link these phrases. Contemporary state-of-the-art Coreference Resolution methods use statistical methods or sets of rules or both, but rarely go beyond the surface of the text. We propose an adaptation of the Stanford “dcoref” Coreference Resolution architecture using computational logic, which will allow an integration of knowledge. We describe a basic rule framework for applying the deterministic architecture of Stanford Sieve which shows that realizing the deterministic part is feasible. Our work is the first step toward making the deterministic architecture nondeterministic. Our vision is that this will allows the new architecture to flexibly integrate semantic knowledge in order to (i) impose constraints on links as well as (ii) generate new candidates for links that cannot be discovered based on shallow semantic knowledge. Our empirical results with our framework suggest, that computational logic provides the desired flexibility, however it also requires significantly more resources. This can be averted by tuning the rules for performance, which unfortunately makes the rules less maintainable and suggests future development of automatic rule-base optimization algorithms.

Keywords: Coreference resolution, Sieve system, Stanford CoreNLP.

List of Abbreviations

ASP	Answer Set Programming
BLANC	BiLateral Assessment of NounPhrase Coreference
CEAF	Constrained Entity Aligned F-measure
CNL	Controlled Natural Language
CR	Coreference Resolution
ID	Identifier
GDT	Generate, Define, and Test
NER	Named Entity Recognition
NLP	Natural Language Processing
NP	Noun Phrases
P	Precision
POS	Part Of Speech
R	Recall

List of Figures

List of Tables

Chapter 1

Introduction

Coreference resolution is the task of determining noun phrases that denote the same real world entity in natural language [29, 14, 21] such noun phrases are called mentions. They can be either nominal (Mr. Smith) or pronominal (she). In the sentence ”He said to the people: ’I need your help’ " the mentions ”he" and ”I" will be connected as they denote the same entity (the speaker), and ”the people" and ”your" will also be connected, as they denote the listener. Nominal coreference is challenging, Noun phrases (NP) can be synonyms, hypernyms, or hyponyms, or they can be linked because of discourse information and background knowledge [22]. Pronominal coreference is more challenging, it requires deep language understanding and use of background knowledge, where linguistic constraints (e.g., agreement of features like gender or number) are not sufficient for connecting the pronoun with the correct entity. Coreference resolution is important for natural language understanding tasks like summarization, question answering, and information extraction [16].

Research in coreference resolution showed that to achieve high quality resolution it is extremely important to use highly precise lexical and syntactic features, and to perform coreference resolution together for several mentions rather than considering only just two mentions [16]. In Coreference resolution, several methods have been proposed, mostly depends on machine learning Denis and Baldridge (2007) describe a supervised statistical approach based on a Maximum Entropy Model which is evaluated using Integer Linear Programming. Culotta, Wick, and Mccallum (2007) use Markov Logic Networks. Supervised machine learning approaches to coreference resolution are more common than unsupervised ones, they require expensive labeled data, and generalize not so well to new words or domains. Unsupervised systems like Haghighi and Klein (2007); Ng (2008); Poon and Domingos (2008) who uses Expectation Maximization, Markov Logic and a nonparametric Bayesian model, respectively are attractive due to the availability of large quantities of unlabeled text. However, unsupervised coreference resolution is much more complicated, making them difficult to be applied to new cases and genres too.

Answer Set Programming (ASP) is a general purpose logic programming formalism oriented towards difficult search problems, it is an evolution of research on the use of nonmonotonic reasoning and knowledge representation, which has been getting increasing attention during the last years [9, 12, 18]. ASP is relatively new, it has been proposed as a programming paradigm in 1999 and has been applied for problems from many areas since then. ASP has proven to be an effective approach for many important computational problems in combinatorial optimization, constraint satisfaction and artificial intelligence. Nevertheless, it has mainly been used by people from academia and has not become a mainstream programming approach yet [15]. One possible explanation for that is that writing an answer-set program is quite different from what developers are used to, developers are used to tools, methods, and methodologies that ease the programming process, however many of these techniques cannot be applied to ASP in a straightforward way [24]. In an ASP logic program we describe (i) a set of potential solutions, (ii) relationships between concepts in the solution, and (iii) constraints on solutions. This logical representation of the problem will be given to ASP solver (a software tool) that will compute solutions that stick to the specified relationships and constraints.

Stanford's Sieve [16] is recently introduced deterministic coreference resolution approach that is now the most successful, and surpassed pure machine learning approaches [20]. It integrates the global information and accurate characteristic of new machine-learning patterns with the transparency and modularity of deterministic, rule-based systems [16]. The Sieve consists of ten passes that will be applied one at a time from the highest to to lowest precision. The output of each pass is a set of clusters (entities) where each cluster contains the mentions that have been linked. Each model processes the output comes from the previous model in deterministic way i.e. links that have been considered can not be reconsidered by any other model.

In this work, we aim to generalize the Sieve architecture using computational logic, to allow usage of knowledge resources to assist coreference resolution, to prevent wrong links between mentions (attribute checks) and to generate candidates for links between mentions using Answer Set Programming (ASP), we represent the Sieve architecture and its individual modules completely in the rule-based ASP formalism.

So far we have created the theoretical framework for this idea, and performed experiments with the deterministic part of a rule-based replica of Stanford dcoref. Our experiences so far show that the idea is feasible however additional tool support for optimizing programs would greatly ease the process of developing such a system. Creating a rule-based system in a rule-based formalism gives us the advantage of making rules more explicit and having the possibility to make the rules nondeterministic.

Chapter 2

Material and Method

2.1 THE Sieve APPROACH

The Sieve technique [16] is a deterministic, non-statistical, and rule based approach for coreference resolution. it starts by detecting the mentions using high recall algorithm, this algorithm picks out all NPs, pronouns, and named entity as candidate mentions, and after that removes from them non-mentions (quantifier expressions, numeric mentions, stop words, etc.). After that the coreference resolution begins where ten models see figure 2.1 will be applied one at a time to link mentions. Each model processes the output of the previous model (clusters) where only first mentions in the textual order in their cluster will be considered. In general mentions that appear earlier in the text are better defined than the ones that come later, they usually have more modifiers and mostly not pronouns. Also as they are closer to the beginning of the text, they have fewer antecedent candidates, and so fewer opportunities to make a mistake in linking [16]. The Sieve ends up with a small post processing step where mentions with singleton clusters (clusters that have one mention) and links obtained through predicate nominative pattern will be removed, this is important to to align Sieve result with the OntoNotes (used corpora) annotation standards.

The Sieve approach is entity-centric, where linking two mentions does not depend only on their features (non-stop words, head word, gender,number ,etc.), but it considers any information about the other mention in their clusters. Each model uses the information about all the mentions in the clusters not only the considered mention and its antecedent candidate in the coreference, that is so important especially for the pronominal coreference (tenth model) with the highest recall and lowest precision, this model enforces agreement constraints between mentions, and can be strictly affected by missing

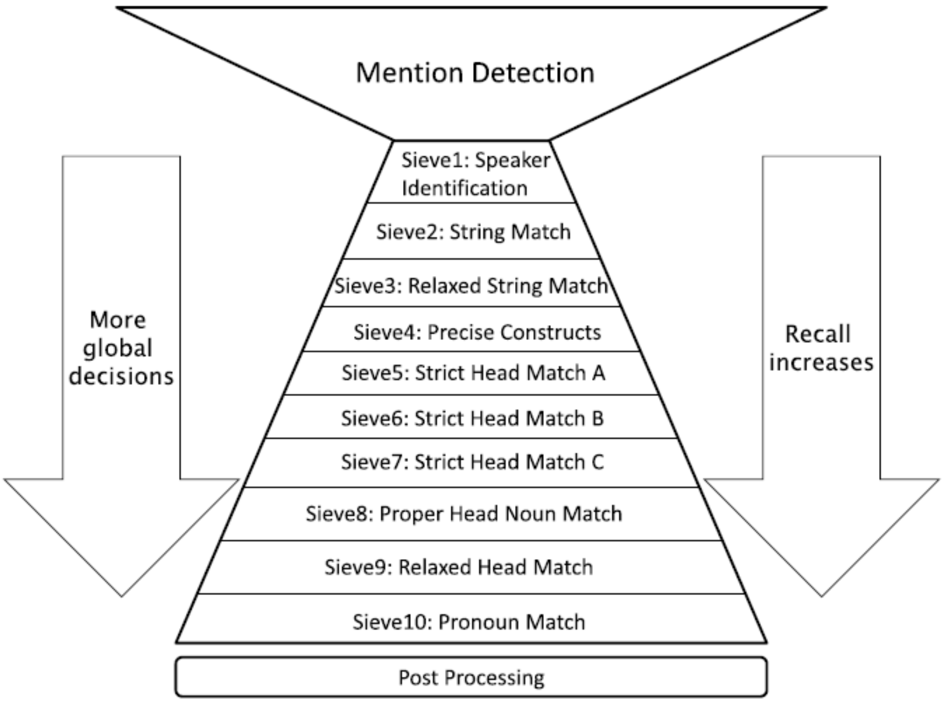


Figure 2.1: Stanford Sieve architecture

attributes [16]. A sample run-through of the Sieve approach [16], consider the following sequence of sentences.

Jack is a painter 2) He painted a new picture.
3) A women was looking to the picture. 4) “It is my favorite, ”Jack said to her.

As we mentioned, the Sieve starts by detecting mentions and assigns each one to a cluster, superscript and subscript are used to mark cluster ID and mention ID.

$$[Jack]^1_1\ is\ [a\ painter]^2_2.\ [He]^3_3\ painted\ [a\ new\ picture]^4_4.$$

$$[A\ woman]^5_5\ was\ looking\ to\ [the\ picture]^6_6.\ ``[It]^7_7\ is\ [[my]^9_9\ favorite]^8_8\ ,''\ [Jack]^{10}_{10}\ said\ to[her]^{11}_{11}.$$

the first model *Speaker Identification* with the highest precision matches speakers to compatible pronouns that appear in a quotation. So the mentions *my*, and the speaker *Jack* will be linked in this model, and their clusters will be merged in one cluster number *nine*

$$[Jack]^1_1\ is\ [a\ painter]^2_2.\ [He]^3_3\ painted\ [a\ new\ picture]^4_4.$$

$$[A\ woman]_5^5\ was\ looking\ to\ [the\ picture]_6^6.\ \text{``}[It]_7^7\ is\ [[my]_9^9\ favorite]_8^8\ ,\text{''}\ [Jack]_{10}^9\ said\ to[her]_{11}^{11}.$$

The second model *String Match* links a nominal mention with its antecedent if they have the same extent text, so the tenth mention, *Jack* will be linked to the first mention *Jack*, and their clusters will be merged

$$[Jack]_1^1\ is\ [a\ painter]_2^2.\ [He]_3^3\ painted\ [a\ new\ picture]_4^4.$$

$$[A\ woman]_5^5\ was\ looking\ to\ [the\ picture]_6^6.\ \text{``}[It]_7^7\ is\ [[my]_9^1\ favorite]_8^8\ ,\text{''}\ [Jack]_{10}^1\ said\ to[her]_{11}^{11}.$$

The *relaxed string match* model links two nominal mentions satisfying a more relaxed string matching constraints than exact match, there are no such mentions, so no changes in the clusters. In the precise constructs model, several syntactic constructs (appositive relations, predicate nominatives, role appositive, etc.) have been used to link mentions. Two predicate nominative relations exist in this example, in the first sentence the mentions *Jack* and *a painter* will be linked, also the mentions *it* and *my favorite* in the fourth sentence.

$$[Jack]_1^1\ is\ [a\ painter]_2^1.\ [He]_3^3\ painted\ [a\ new\ picture]_4^4.$$

$$[A\ woman]_5^5\ was\ looking\ to\ [the\ picture]_6^6.\ \text{``}[It]_7^7\ is\ [[my]_9^1\ favorite]_8^7\ ,\text{''}\ [Jack]_{10}^1\ said\ to[her]_{11}^{11}.$$

The next four models (5,6,7, and 8) link mentions with the same head word, and satisfying several constraints that differ between them. The mentions *a new picture* and *the picture* are linked in such models.

$$[Jack]_1^1\ is\ [a\ painter]_2^1.\ [He]_3^3\ painted\ [a\ new\ picture]_4^4.$$

$$[A\ woman]_5^5\ was\ looking\ to\ [the\ picture]_6^4.\ \text{``}[It]_7^7\ is\ [[my]_9^1\ favorite]_8^7\ ,\text{''}\ [Jack]_{10}^1\ said\ to[her]_{11}^{11}.$$

The tenth model is *Pronominal Coreference Resolution*, this model uses a standard approach for many years : enforcing attribute agreement constraints such as gender, and number between the coreferent mentions, several links can be made according to this model

- *he* and first mention *Jack*.
- *it* and *the picture*.
- *her* and *a woman*.

$$[Jack]^1_1 \; is \; [a \; painter]^1_2. \; [He]^1_3 \; painted \; [a \; new \; picture]^4_4.$$

$$[A \; woman]^5_5 \; was \; looking \; to \; [the \; picture]^4_6. \; \text{``}[It]^4_7 \; is \; [[my]^1_9 \; favorite]^7_8 \; ,'' \; [Jack]^1_{10} \; said \; to[her]^5_{11}.$$

At last post-processing step comes, where singleton clusters, and links made using predicate nominative relation will be removed.

$$[Jack]^1_1 \; is \; a \; painter. \; [He]^1_3 \; painted \; [a \; new \; picture]^4_4.$$

$$[A \; woman]^5_5 \; was \; looking \; to \; [the \; picture]^4_6. \; \text{``}[It]^4_7 \; is \; [my]^1_9 \; favorite \; ,'' \; [Jack]^1_{10} \; said \; to[her]^5_{11}.$$

So we get the following sets of mentions or chains (also called entities) :

- {1; 3; 9; 10} (Jack, He, my, Jack);
- {4; 6; 7} (a new picture, the picture, It);
- {5; 11} (A woman, her).

2.2 ANSWER SET PROGRAMMING

ASP is a form of declarative problem solving approach, initially oriented towards modeling problems in the area of knowledge representation and reasoning. It is based on the stable model (answer set) semantics of logic programming. It is a Balance between expressivity, ease of use, and computational effectiveness [4, 2]. A common methodology to solve a problem in ASP is *GENERATE*, *DEFINE*, and *TEST (GDT)*, which is sometimes also called *Guess and Check* .

The GENERATE part defines the search space, a large collection of answer sets that could be seen as potential solutions. The TEST part prunes all bad potential solutions. The DEFINE section expresses additional concepts, and connects the GENERATE and TEST parts [3, 8].

ASP program is a set of rules, some rules are similar to traditional Prolog rules, for instance the program

$$p \leftarrow q.$$
$$q \leftarrow not\ r.$$

consists of such rules, and has one stable model, which consists of *p* and *q*. In addition to Prolog rule ASP program consists of other kind of rules, *choice rules*, and *constraints*. choice rules are the main elements of the GENERATE part of the program that generate the search space, for example

$$\{s, t\} \leftarrow p.$$

It means that if *p* is included in the stable model, the answer sets of this one rule program are arbitrary subsets of the atoms *s*, *t*. Constraints form the TEST part of the program, constraints define the conditions that must be met, accordingly several candidate solutions will be eliminated. Rules with an empty head represent the constraints. For instance, the constraint

$$\leftarrow p, not\ q.$$

eliminates the answer sets that include *p* and do not include *q*.

ASP Programs is usually processed in two steps. The First step is *grounding* where the program with variables is replaced by an equivalent program without variables. The second step is *solving* this propositional program by a backtracking search algorithm that finds one or more of its answer sets or determines that no answer sets exist. The currently used software tools in each step referred to as grounder and solver, respectively, have already reached the level of performance that makes it possible to use them successfully with programs arising from problems of practical importance [24, 5, 7].

2.2.1 Answer Set Programming For Natural Languages Processing

The GDT methodology is similar to *ambiguities* in natural language where certain levels of linguistic representation contain ambiguities that generate a set of possible decisions, and other levels of representation test if linguistic constraints are satisfied. These generated possibilities and tests can be connected by several layers of linguistic representation. For example, in the sentence ”He deposited money at the bank." assigning a meaning to the word ”bank" generates the possibilities that bank is a money institute or a river bank or a bench to sit on. Semantic modules, e.g., FrameNet can then define a more holistic representation of the sentence that captures more of the overall meaning than a syntactic representation. In our example such a holistic representation can contain the information that the ”money institute" decision makes the sentence a common event in daily life, whereas the ”river bank" or ”bench" decisions means that the sentence contains an unusual (possibly criminal) activity. As a consequence a test module that selects if the text before hints at criminal activity can eliminate at least one of the decisions such that only valid results remain. ASP is the ideal tool for addressing ambiguity in natural language because ASP is designed to perform exactly such reasoning, and to perform it in an efficient way.

For example, in [11] they introduced a Controlled Natural Language (CNL) which is a formal language but with a look of a natural language for biomedical queries to facilitate access to biomedical ontologies. CNL is similar to natural languages with a restricted grammar and vocabulary, to overcome the ambiguity of natural languages, so it can be easily converted to ASP program. Automated reasoners in ASP can then be used to find answers to queries expressed in a CNL.

ASP community has created several tools for performing reasoning with hybrid knowledge sources within ASP and those software tools — in particular clingo [13], and wasp [1] are tuned for efficient reasoning and freely available to use in the our project.

2.3 STANFORD CORENLP TOOLKIT

We will focus only on the second stage of the Sieve (coreference) and reuse existing methods for mention detection, *Stanford CoreNLP toolkit* [20] have been used, it is extensible pipeline that provides core natural language analysis. This toolkit is quite widely used, both in the research NLP community, and also among commercial and government users of open source NLP technology [20].

Mentions, that have been detected, and information about them have been represented as *facts* in ASP, for each mention we have the mention text *M*, Identifier *ID*, sentence number *SN* (Number goes from 0 upwards), animacy attribute (animate, inanimate) *ANIM*, word index of the mention's first word in its sentence *SI*, word Index of its last word *EI*, gender attribute (male, female, neutral) *G*, head word of the mention *HW*, type of the mention (proper, nominal, pronominal, ..) *T*, person attribute *P* just for pronominal mentions (for example my mention gets I person attribute and so on...), named entity recognition tag *NER* (names of persons, organizations, locations, expressions of times, monetary values,...), number attribute (Singular, plural) *ISPR*. If Stanford is not able to determine the value of the attribute, it will get *UNKNOWN* value.

$$mention(M, ID, SN, ANIM, SI, EI, G, HW, T, P, NER, ISPR).$$
$$mention("marmara\ university", 1, 0, "INANIMATE", 5, 7,$$
$$"NEUTRAL", "university", "PROPER", "UNKNOWN",$$
$$"ORGANIZATION", "UNKNOWN").$$

Stanford assigns values to the mention's attribute as follow

1. Number: are assigned according to: (a) pronouns' list; (b) NER labels: named entity mentions are singular except for organizations(singular or plural); c) part of speech (POS) tags: NN* tags are singular except for NN*S tags are plural; and (d) Bergsma and Lin (2006) static dictionary [16].

$$attribute("number").$$

2. Gender: are assigned based on Bergsma and Lin (2006), and Ji and Lin (2009) static lexicons.

$$attribute("gender").$$

3. Person: are assigned only for pronominal mentions. this constraint is not required when linking pronouns.

$$attribute("person").$$

4. Animacy: are assigned using: (a) pronouns' list; (b) NER labels: where mentions with PERSON NER label are animate while LOCATION mentions are not; and (c)the Web (Ji and Lin 2009) dictionary bootstrapped [16].

$$attribute("animacy").$$

5. NER label :are assigned using Stanford NER [16].

$$attribute("ner").$$

Words of each mention are also important as input to the Sieve, so the Mention with identifier *ID* has a word *W*, this word has an identifier *WS*, type (stop, nonstop) *WT*, and sequence number in whole text *WSS*.

$$word(ID, W, WS, WT, WSS).$$

$$word(1, "university", 1, "nonstop", 2).$$

Also mention's Modifiers, mention with the *ID* has a modifier *MM*, with an identifier *S*, part of speech POS tag *T*.

$$mod(ID, S, MM, T).$$

$$mod(1, 1, "marmara", "NNP").$$

NER tag for the modifier, the modifier of the mention *ID* with the identifier *S* has NER tag *NER*.

$$loctagofmod(ID, S, NER).$$

$$loctagofmod(2, 2, "ORGANIZATION").$$

Part of speech POS tag of the mention's head word *POS*

$$headwordtype(ID, POS).$$

$$headwordtype(1, "nn").$$

The mention *ID* represents a name of state or abbreviation of state's name, for example mention with the text *California*, or its abbreviation *CA*.

$$stateorabb(ID).$$

We add more facts such dependency parsing information obtained from Stanford, for example *Predicate nominative* i.e. the two mentions are in a copulative subject–object relation, mentions *Marmara University* and *one of the oldest educational institutions* have such relation

$$[Marmara\ University]\ is\ [one\ of\ the\ oldest\ educational\ institutions].$$

$$predicatenominative(6, 5).$$

2.4 ASP ENCODING OF THE Sieve SYSTEM

The facts of the passes, and their sequences. The atom *nextStage(Pass1,Pass2)* defines the order of the passes, the source pass *Pass2* and the result *Pass1*. The atom *activatePass(Pass)* activates passes

$$nextStage(0, 1). \quad activatePass(1).$$

$$nextStage(1, 2). \quad activatePass(2).$$

$$nextStage(2, 3). \quad activatePass(3).$$

$$nextStage(3, 41). \quad activatePass(41).$$

$$nextStage(41, 42). \quad activatePass(42).$$

$$nextStage(42, 43). \quad activatePass(43).$$

$$nextStage(43, 44). \quad activatePass(44).$$

$$nextStage(44, 45). \quad activatePass(45).$$

$$nextStage(45, 46). \quad activatePass(46).$$

$$nextStage(46, 4). \quad activatePass(4).$$

$$nextStage(4, 5). \quad activatePass(5).$$

$$nextStage(5, 6). \quad activatePass(6).$$

$$nextStage(6, 7). \quad activatePass(7).$$

$$nextStage(7, 8). \quad activatePass(8).$$

$$nextStage(8, 9). \quad activatePass(9).$$

$$nextStage(9, 10). \quad activatePass(10).$$

There are several facts and rules have been used in the implementation of the Stanford Sieve. First person pronouns as ASP facts.

$$firstpersonpronoun("i").$$

$$firstpersonpronoun("me").$$

$$firstpersonpronoun("myself").$$

$$firstpersonpronoun("mine").$$

$$firstpersonpronoun("my").$$

$$firstpersonpronoun("we").$$

$$firstpersonpronoun("us").$$

$$firstpersonpronoun("ourself").$$

$$firstpersonpronoun("ourselves").$$

$$firstpersonpronoun("ours").$$

$$firstpersonpronoun("our").$$

Also second person pronouns.

$$secondpersonpronoun("you").$$

$$secondpersonpronoun("yourself").$$

$$secondpersonpronoun("yours").$$

$$secondpersonpronoun("your").$$

$$secondpersonpronoun("yourselves").$$

Location facts

$$locationmodifier("east").$$

$$locationmodifier("west").$$

$$locationmodifier("north").$$

$$locationmodifier("south").$$

$$locationmodifier("eastern").$$

$$locationmodifier("western").$$

$$locationmodifier("northern").$$

$$locationmodifier("southern").$$

$$locationmodifier("upper").$$

$$locationmodifier("lower").$$

Mention's head word

$$hwmen(ID, HW) \leftarrow mention(M, ID, SN, ANIM, SI, EI, G, HW, T, P, NER, ISPR).$$

$$\begin{aligned} hwmen(1, "university") \leftarrow mention(&"marmara\ university", 1, 0, "INANIMATE",\\ &5, 7, "NEUTRAL", "university", "PROPER", "UNKNOWN",\\ &"ORGANIZATION", "UNKNOWN"). \end{aligned}$$

Cluster's head words set includes the head word of every mention in this cluster, Cluster *C* has a head word *HW* in the pass *P*.

$$hwsc(C, HW, P) \leftarrow mention(M, ID, SN, ANIM, SI, EI, G, HW, T, P, NER, ISPR),$$

$$incluster(ID, P, C).$$

$$hwsc(4, "university", 2) \leftarrow mention("marmara\ university", 1, 0, "INANIMATE",$$
$$5, 7, "NEUTRAL", "university", "PROPER", "UNKNOWN",$$
$$"ORGANIZATION", "UNKNOWN"), incluster(1, 2, 4).$$

Clusters of each pass *Pass*.

$$cluster(C, Pass) \leftarrow incluster(ID, Pass, C).$$

Mentions in the same clusters in a pass *Pass*.

$$samecluster(ID1, ID2, Pass) \leftarrow incluster(ID1, Pass, C),$$
$$incluster(ID2, Pass, C), ID1 != ID2.$$

Pronominal mentions, mention's type *T* is *PRONOMINAL*

$$pronominal(ID) \leftarrow mention(M, ID, SN, ANIM, SI, EI, G, HW,$$
$$T, P, NER, ISPR), T == "PRONOMINAL".$$

Nominal mentions, mention's type *T* equlas one of the values *{NOMINAL;PROPER;LIST}*

$$nominal(ID) \leftarrow mention(M, ID, SN, ANIM, SI, EI, G, HW, T, P,$$
$$NER, ISPR), T == ("NOMINAL"; "PROPER"; "LIST").$$

Links in all passes are symmetric and transitive

$$link(ID2, ID1, Pass) :- link(ID1, ID2, Pass).$$
$$link(ID1, ID3, Pass) :- link(ID1, ID2, Pass), link(ID2, ID3, Pass),$$
$$ID1 != ID2, ID2 != ID3, ID1 != ID3.$$

2.4.1 Mention Selection In a Given Sieve

Before the Sieve starts, we have a cluster for each mention, passing through the models links are detected, and clusters are merged.The atom *incluster(ID, P,C)* indicates the cluster *C* of the mention *ID* after a pass *P*. Mention with *ID* 50 is in cluster 50 in pass zero i.e. before Sieve system starts

$$incluster(50, 0, 50).$$

the variable P in the atom $incluster(ID, P, C)$ goes from zero to ten, here the mention has changed its cluster after pass 6 to 34 .

$$incluster(50, 6, 34).$$

Only mentions with the smallest *ID* in its clusters will be considered in the resolution, ASP aggregate *min* are used to get those mentions for every clusters, so Mention *Y* is the first mention in the cluster *C* in Pass *P*.

$$\mathit{firstmenofclu}(Y, P, C) \leftarrow Y = \#\mathit{min}\{ID : \mathit{incluster}(ID, P, C), \mathit{incluster}(Y, P, C)\}.$$

Search space have been pruned using a simple model of discourse salience [16]. Some mentions even though they are first in their clusters will not be considered in coreference.

(a) Are or start with indefinite pronouns, the first word *W* (its identifier is zero) of the mention *ID* is an indefinite pronoun.

$$\begin{aligned} \mathit{indefinitepronounstrw}(ID) \leftarrow {} & \mathit{word}(ID, W, 0, WT, WS), \\ & W == (\text{"}\mathit{another}\text{"}; \text{"}\mathit{anybody}\text{"}; \text{"}\mathit{anyone}\text{"}; \text{"}\mathit{anything}\text{"}; \\ & \text{"}\mathit{each}\text{"}; \text{"}\mathit{any}\text{"}; \text{"}\mathit{less}\text{"}; \text{"}\mathit{somebody}\text{"}; \text{"}\mathit{somebody}\text{"}; \\ & \text{"}\mathit{either}\text{"}; \text{"}\mathit{enough}\text{"}; \text{"}\mathit{everybody}\text{"}; \text{"}\mathit{everyone}\text{""}; \\ & \text{"}\mathit{neither}\text{"}; \text{"}\mathit{nobody}\text{"}; \text{"}\mathit{nothing}\text{"}; \text{"}\mathit{one}\text{"}; \text{"}\mathit{other}\text{"}; \text{"}\mathit{plenty}\text{"}; \\ & \text{"}\mathit{something}\text{"}; \text{"}\mathit{both}\text{"}; \text{"}\mathit{few}\text{"}; \text{"}\mathit{fewer}\text{"}; \text{"}\mathit{many}\text{"}; \text{"}\mathit{others}\text{"}; \\ & \text{"}\mathit{more}\text{"}; \text{"}\mathit{most}\text{"}; \text{"}\mathit{none}\text{"}; \text{"}\mathit{some}\text{"}; \text{"}\mathit{such}\text{"}; \text{"}\mathit{little}\text{"}; \text{"}\mathit{much}\text{"}; \\ & \text{"}\mathit{someone}\text{"}; \text{"}\mathit{several}\text{"}; \text{"}\mathit{all}\text{"}; \text{"}\mathit{everything}\text{"}; \text{"}\mathit{somebody}\text{"}). \end{aligned}$$

For the *"no one"* indefinite pronoun, both the first word *W1* and the second word *W2* of the mention *ID* have to be checked.

$$\begin{aligned} \mathit{indefinitepronounstrw}(ID) \leftarrow {} & \mathit{word}(ID, W1, 0, WT1, WS1), \\ & \mathit{word}(ID, W2, 1, WT2, WS2), W1 == \text{"}\mathit{no}\text{"}, W2 == \text{"}\mathit{one}\text{"}. \end{aligned}$$

(b) Start with indefinite articles, the first word *W* of the mention *ID* is an indefinite article

$$\mathit{indefinitearticlestr}(ID) \leftarrow \mathit{word}(ID, W, 0, WT, WS), W == (\text{"}\mathit{an}\text{"}; \text{"}\mathit{a}\text{"}).$$

(c) Are bare plurals like (restaurants, dogs. . . .)[16].

$$\mathit{generic}(ID).$$

Pruning search for the mention ID for one of the previously mentioned cases

$$prunemen(ID) \leftarrow indefinitepronounstrw(ID); indefinitearticlestr(ID); gneric(ID).$$

The pruning will be applied in all passes except for Exact String Match pass, which links mentions with the same text[16].

2.4.2 Feature Sharing In The Entity-Centric Model

Using all the information about every mention in the clusters of the mention to be considered and its antecedent candidate not only their local features is so important in linking those two mentions especially in pronominal coreference. So for each cluster all its mentions' attributes (number,person,NER, gender, animacy) participate in the linking decision. For example if those two mentions are in the same cluster

$$mention(\text{"}a\ group\ of\ students\text{"}, 58, 16, \text{"}INANIMATE\text{"}, 2, 6, \text{"}NEUTRAL\text{"},$$
$$\text{"}group\text{"}, \text{"}NOMINAL\text{"}, \text{"}UNKNOWN\text{"}, \text{"}O\text{"}, \text{"}SINGULAR\text{"}).$$
$$mention(\text{"}fivestudents\text{"}, 61, 17, \text{"}ANIMATE\text{"}, 4, 6, \text{"}UNKNOWN\text{"}, \text{"}students\text{"},$$
$$\text{"}NOMINAL\text{"}, \text{"}UNKNOWN\text{"}, \text{"}O\text{"}, \text{"}PLURAL\text{"}).$$

Attributes' sets of this cluster become

1. Number attribute *{SINGULAR, PLURAL}*
2. Ner attribute *{O}*
3. Person attribute *{ UNKNOWN }*
4. Gender attribute *{ NEUTRAL, UNKNOWN }*
5. Animacy attribute *{ INANIMATE, ANIMATE }*

Therefore, this cluster can be merged with both inanimate and animate pronouns. Attributes rules for the mentions

$$attributeofmention(ID, \text{"}number\text{"}, ISPR) \leftarrow mention(M, ID, SN, ANIM, SI, EI,$$

$$G, HW, T, P, NER, ISPR).$$

$$attributeofmention(ID, "animacy", ANIM) \leftarrow mention(M, ID, SN, ANIM, SI, EI, G, HW, T, P, NER, ISPR).$$

$$attributeofmention(ID, "gender", G) \leftarrow mention(M, ID, SN, ANIM, SI, EI, G, HW, T, P, NER, ISPR).$$

$$attributeofmention(ID, "person", P) \leftarrow mention(M, ID, SN, ANIM, SI, EI, G, HW, T, P, NER, ISPR).$$

$$attributeofmention(ID, "ner", "UNKNOWN") \leftarrow mention(M, ID, SN, ANIM, SI, EI, G, HW, T, P, NER, ISPR), NER == ("O"; "MISC").$$

$$attributeofmention(ID, "ner", NER) \leftarrow mention(M, ID, SN, ANIM, SI, EI, G, HW, T, P, NER, ISPR), NER! = "O", NER! = "MISC".$$

Cluster's attribute values are the attribute's value of its mentions

$$attributeofc(Cluster, Value, attribute\_Type, Pass) \leftarrow incluster(MentionID, Pass, Cluster), attributeofmention(MentionID, attribute\_Type, Value), attribute\_agreement\_for\_stage(Pass).$$

The two clusters can have agreement in an attribute in one of the two cases

- The attribute *att_Type* has *UNKNOWN* value in its value list in both clusters *C1* and *C2* in pass ***P***

$$attributeagreement(C1, C2, att\_Type, P) \leftarrow V\_1 == "UNKNOWN", V\_2 == "UNKNOWN", attributeofc(C1, V\_1, att\_Type, P), attributeofc(C2, V\_2, att\_Type, P), C1! = C2, attribute\_agreement\_for\_stage(P).$$

- If not, the values of each attribute except for the *UNKNOWN* value for one cluster must be included in the value set of the other cluster (inclusion between attribute value sets of the clusters). We are checking if there is at least one value for an attribute that is in the attribute set of one cluster, and not in the other cluster attribute set. Counting the attribute's values that are not *UNKNOWN* in each cluster, cluster *C* has *X* values for the attribute *att_Type* in pass *P*

$$\begin{aligned} noofcatt(C, X, att\_Type, P) \leftarrow X = \#count\{&C, V, att\_Type, \\ &P : attributeofc(C, V, att\_Type, P), \\ &V! = \text{"}UNKNOWN\text{"}\}, cluster(C, P), attribute(att\_Type), \\ &attribute\_agreement\_for\_stage(P). \end{aligned}$$

For pairs of distinct clusters and attribute value that occurs in both clusters, clusters *C1*, *C2* have the same value *Value* which is not *UNKNOWN* for the attribute *att_Type* in the pass *P*

$$\begin{aligned} att\_has\_partner(C1, C2, V, P, att\_Type) \leftarrow &attribute\_agreement\_for\_stage(P), \\ &attributeofc(C1, V, att\_Type, P), \\ &attributeofc(C2, V, att\_Type, P), \\ &Value! = \text{"}UNKNOWN\text{"}, cluster(C1, P), \\ &cluster(C2, P), C1! = C2. \end{aligned}$$

Attribute *att_Type* has a value *Value* which is not *UNKNOWN* in cluster *C1*, that is not in cluster *C2*

$$\begin{aligned} att\_has\_no\_partner\_w(C1, C2, V, P, att\_Type) \leftarrow &attribute\_agreement\_for\_stage(P), \\ &attributeofc(C1, V, att\_Type, P), \\ &not\ att\_has\_partner(C1, C2, V, P, \\ &att\_Type), \\ &cluster(C1, P), cluster(C2, P), \\ &C1! = C2, V! = \text{"}UNKNOWN\text{"}. \end{aligned}$$

There is an attribute *att_Type* that has value in *C1* but not in *C2* in pass *P*

$$\begin{aligned} att\_has\_no\_partner(C1, C2, P, att\_Type) \leftarrow &attribute\_agreement\_for\_stage(P), \\ &att\_has\_no\_partner\_w(C1, C2, \_, P, att\_Type), \\ &cluster(C1, P), cluster(C2, P), C1! = C2. \end{aligned}$$

Cluster must have at least one attribute value, except for *UNKNOWN*, *C* have at least one value for the attribute *att_Type* in pass *P*.

$$cluster\_has\_att(C, att\_Type, P) \leftarrow attribute\_agreement\_for\_stage(P),$$

$$noofcatt(C, X, att\_Type, P), X > 0.$$

For the attribute *att_Type*, its values in *C2* are also in *C1* in pass *P*

$$\begin{aligned} attributeagreement(C1, C2, att\_Type, P) \leftarrow\ & not\ att\_has\_no\_partner(C2, C1, P, att\_Type), \\ & cluster\_has\_att(C2, att\_Type, P), \\ & cluster(C1, P), attribute(att\_Type), \\ & attribute\_agreement\_for\_stage(P), \\ & cluster(C2, P), C1! = C2. \end{aligned}$$

As attribute agreement will be used in several passes, this rule is used to activate attribute agreement for those passes

$$attribute\_agreement\_for\_stage(P).$$

2.4.3 Coreference Resolution Sieves

Ten independent models will be applied in sequence

Model 1: Speaker Identification.

In this pass the speakers will be linked to pronouns using the method in Baldwin (1995, 1997) [16]. In non–conversational text, it looks for for the subject of reporting verbs (tell,say...) in the sentence of the quotation or its adjacent sentences. In conversational text, speakers are available in the data sets. The detected speaker relations by Stanford are represented as ASP facts that specify that a mention *ID*, has a speaker mention *SID*

$$speakerofmen(ID, SID).$$

The extracted speaker facts then can be used to implement the following [16]:

1. I pronouns that have the same speaker are coreferent [16]. For example *"[I] am studying at Marmara University, and [I] am working,"[she] said.* The two *(I)s* mentions will be linked, they have the same speaker *she* mention, so the mentions that are first person pronouns *first-personpronoun(M1)*, with singular number attribute *attributeofmention(ID1,"number", "SINGULAR")*, and have same speaker will be linked.

$$link(ID1, SID1, 1) \leftarrow mention(M1, ID1, SN1, ANIM1, SI1, EI1, G1,$$

$$HW1, MT1, P1, NER1, ISPR1), \mathit{firstpersonpronoun}(M1),$$
$$\mathit{attributeofmention}(ID1, \text{"}\mathit{number}\text{"}, \text{"}SINGULAR\text{"}),$$
$$\mathit{speakerofmen}(ID1, SID1), ID1 < SID1, \mathit{not\ prunemen}(SID1).$$

2. The speaker and (I)s in her text are coreferent. The *(I)s* mentions and the *she* mention in the previous example are linked .

$$\begin{aligned} \mathit{link}(ID1, SID1, 1) \leftarrow\ & \mathit{mention}(M1, ID1, SN1, ANIM1, SI1, EI1, G1, \\ & HW1, MT1, P1, NER1, ISPR1), \mathit{firstpersonpronoun}(M1), \\ & \mathit{attributeofmention}(ID1, \text{"}\mathit{number}\text{"}, \text{"}SINGULAR\text{"}), \\ & \mathit{speakerofmen}(ID1, SID1), ID1 < SID1, \mathit{not\ prunemen}(SID1). \end{aligned}$$

3. You pronouns that have the same speaker are coreferent . *"[I] invited [you], and [you] did not come,"[he] said.* The two *(you)s* mentions will be linked, they have the same speaker *he* mention, so the mentions that are second person pronouns *secondpersonpronoun(M1)*, and have the same speaker are linked.

$$\begin{aligned} \mathit{link}(ID1, ID2, 1) \leftarrow\ & \mathit{mention}(M1, ID1, SN1, ANIM1, SI1, EI1, G1, \\ & HW1, MT1, P1, NER1, ISPR1), \mathit{secondpersonpronoun}(M1), \\ & \mathit{mention}(M2, ID2, SN2, ANIM2, SI2, EI2, G2, \\ & HW2, MT2, P2, NER2, ISPR2), \mathit{secondpersonpronoun}(M2), \\ & \mathit{speakerofmen}(ID1, SID1), \mathit{speakerofmen}(ID2, SID2), \\ & SID1 == SID2, ID1 != ID2, ID1 < ID2. \end{aligned}$$

4. Link mentions that are speakers, and have the same text

$$\begin{aligned} \mathit{speaker}(SID) \leftarrow\ & \mathit{speakerofmen}(ID, SID). \\ \mathit{mentiontext}(ID1, M1) \leftarrow\ & \mathit{mention}(M1, ID1, SN1, ANIM1, SI1, EI1, \\ & G1, HW1, MT1, P1, NER1, ISPR1). \\ \mathit{link}(SID1, SID2, 1) \leftarrow\ & \mathit{speaker}(SID1), \mathit{speaker}(SID2), \\ & \mathit{mentiontext}(SID1, M1), \mathit{mentiontext}(SID2, M2), \\ & M1 == M2, SID1 != SID2, \\ & \mathit{nominal}(SID1), \mathit{nominal}(SID2). \end{aligned}$$

5. Link nominal speaker mentions with mentions that have same text

$$\begin{aligned} link(SID, ID1, 1) \leftarrow\ & speaker(SID), mentiontext(ID1, M1), \\ & mentiontext(SID, M1), nominal(SID). \end{aligned}$$

Model 2: Exact Match.

Two nominal mentions with the same extent text including modifiers and determiners are linked in this model [16]. For example

$$\begin{aligned} & mention(\text{"}the\ sony\ product\text{"}, 14, 1, \text{"}INANIMATE\text{"}, 6, 9, \text{"}NEUTRAL\text{"}, \\ & \quad \text{"}product\text{"}, \text{"}NOMINAL\text{"}, \text{"}UNKNOWN\text{"}, \text{"}O\text{"}, \text{"}SINGULAR\text{"}). \\ & mention(\text{"}the\ sony\ product\text{"}, 20, 3, \text{"}INANIMATE\text{"}, 8, 11, \text{"}NEUTRAL\text{"}, \\ & \quad \text{"}product\text{"}, \text{"}NOMINAL\text{"}, \text{"}UNKNOWN\text{"}, \text{"}O\text{"}, \text{"}SINGULAR\text{"}). \end{aligned}$$

Nominal mentions with the same text *M* will be linked

$$\begin{aligned} link(ID1, ID2, 2) \leftarrow\ & mention(M, ID1, SN1, ANIM1, SI1, EI1, G1, \\ & HW1, T1, P1, NER1, ISPR1), ID1 < ID2, \\ & mention(M, ID2, SN2, ANIM2, SI2, EI2, G2, \\ & HW2, T2, P2, NER2, ISPR2), nominal(ID1), nominal(ID2). \end{aligned}$$

Model 3: Relaxed String Match.

Nominal mentions are linked if their texts that have been gotten after deleting the words come after their head words (e.g. relative clauses, participial post modifiers,..) are the same [16]. For example

$$\begin{aligned} & mention(\text{"}the\ man,\ who\ won\ the\ contest\text{"}, 4, 0, \text{"}ANIMATE\text{"}, 4, 11, \\ & \quad \text{"}MALE\text{"}, \text{"}man\text{"}, \text{"}NOMINAL\text{"}, \text{"}UNKNOWN\text{"}, \text{"}O\text{"}, \text{"}SINGULAR\text{"}). \\ & mention(\text{"}the\ man\text{"}, 5, 0, \text{"}ANIMATE\text{"}, 4, 6, \text{"}MALE\text{"}, \\ & \quad \text{"}man\text{"}, \text{"}NOMINAL\text{"}, \text{"}UNKNOWN\text{"}, \text{"}O\text{"}, \text{"}SINGULAR\text{"}). \end{aligned}$$

Text of the mention *ID* after dropping words following its head word *Mnew*

$$removephraseafterheadmen(ID, Mnew).$$

For the above two mentions we get

$$removephraseafterheadmen(4, \text{"}the\ man\text{"}).$$

$$removephraseafterheadmen(5, \text{"the man"}).$$

Link the mentions

$$\begin{aligned}link(ID1, ID2, 3) \leftarrow\ & removephraseafterheadmen(ID1, M),\\ & removephraseafterheadmen(ID2, M), nominal(ID1),\\ & nominal(ID2), firstmenofclu(ID2, 2, C2),\\ & ID1 < ID2, notprune(ID2, 2, C2).\end{aligned}$$

Model 4: Precise Constructs

Two mentions are linked, if any of the following situations occurred[16], each condition can be considered as sub model of the main model (4th model). As the third argument in the atom *link(ID1,ID2,Pass)* indicates the model in which the mentions have been linked, in this model we indicate both the main model with number 4, and the sub model number which goes from (1 to 6)

$$\begin{aligned}& link(ID1, ID2, 4SubModelNO).\\ & link(5, 8, 43).\end{aligned}$$

1. Appositive: the two nominal mentions have an appositive relation. Stanford have been used to detect appositive using Haghighi and Klein (2009) definition: third child of a parent NP whose expansion begins with (NP, NP), with no conjunction in the expansion [16]. For example, ''[Istanbul city], [the capital of Turkey], is so beautiful. ''. The mentions have appositive relation

$$\begin{aligned}& mention(\text{"istanbul city, the capitaof turkey"}, 9, 1, \text{"INANIMATE"}, 0, 2,\\ &\qquad \text{"NEUTRAL"}, \text{"city"}, \text{"PROPER"}, \text{"UNKNOWN"}, \text{"LOCATION"}, \text{"SINGULAR"}).\\ & mention(\text{"the capital of turkey"}, 10, 1, \text{"INANIMATE"}, 3, 7,\\ &\qquad \text{"NEUTRAL"}, \text{"capital"}, \text{"NOMINAL"}, \text{"UNKNOWN"}, \text{"O"}, \text{"SINGULAR"}).\\ &\qquad appositive(9, 10).\end{aligned}$$

Link mentions if they have appositive relation *appositive(ID1,ID2)*, and there are attribute agreements over all attribute of their clusters.

$$\begin{aligned}& attribute\_agreement\_for\_stage(3).\\ & appositive(ID2, ID1) \leftarrow appositive(ID1, ID2).\end{aligned}$$

$$
\begin{aligned}
appositive(ID1, ID3) \leftarrow\ & appositive(ID1, ID2), appositive(ID2, ID3),\\
& ID1! = ID2, ID2! = ID3, ID1! = ID3.\\
link(ID1, ID2, 41) \leftarrow\ & appositive(ID1, ID2), firstmenofclu(ID2, 3, C2),\\
& ID1 < ID2, notprune(ID2, 3, C2), notsamecluster(ID1, ID2, 3),\\
& attributeagreement(C1, C2, "number", 3), incluster(ID1, 3, C1),\\
& attributeagreement(C1, C2, "animacy", 3),\\
& attributeagreement(C1, C2, "gender", 3),\\
& attributeagreement(C1, C2, "ner", 3),\\
& attributeagreement(C1, C2, "person", 3).
\end{aligned}
$$

2. Predicate nominative : the mentions are in a copulative subject–object relation [16]. For example *“[Marmara University] is [one of the oldest educational institutions].”*, the mentions

$$
\begin{aligned}
& mention("marmara\ university", 4, 1, "INANIMATE", 0, 2, "NEUTRAL",\\
& \quad "university", "PROPER", "UNKNOWN", "ORGANIZATION", "UNKNOWN").\\
& mention("one\ of\ the\ oldest\ educational\ institutions", 6, 1, "INANIMATE", 4, 10,\\
& \quad "UNKNOWN", "one", "PROPER", "UNKNOWN", "NUMBER", "SINGULAR").\\
& \quad predicatenominative(6, 4).
\end{aligned}
$$

Link mentions if they have predicate nominative relation *predicatenominative (ID1,ID2)*, and there are attribute agreements over all attributes of their clusters.

$$
\begin{aligned}
& attribute\_agreement\_for\_stage(41).\\
predicatenominative(ID2, ID1) \leftarrow\ & predicatenominative(ID1, ID2).\\
predicatenominative(ID1, ID3) \leftarrow\ & predicatenominative(ID1, ID2),\\
& predicatenominative(ID2, ID3),\\
& ID1! = ID2, ID2! = ID3, ID1! = ID3.\\
link(ID1, ID2, 42) \leftarrow\ & predicatenominative(ID1, ID2),\\
& firstmenofclu(ID2, 41, C2),\\
& ID1 < ID2, notprune(ID2, 41, C2),\\
& notsamecluster(ID1, ID2, 41),\\
& attributeagreement(C1, C2, "number", 41),\\
& attributeagreement(C1, C2, "animacy", 41),
\end{aligned}
$$

$$attributeagreement(C1, C2, "gender", 41),$$
$$attributeagreement(C1, C2, "ner", 41),$$
$$attributeagreement(C1, C2, "person", 41),$$
$$incluster(ID1, 41, C1).$$

3. Role appositive:the candidate antecedent is a modifier in an NP whose head is the current mention, and its head word is noun [16]. For example, *"[[Architect] Sinan] is considered as the greatest Ottoman architect of the Ottoman Empire's Architectural heritage."*, and the mentions

$$mention("architect\ sinan", 64, 18, "ANIMATE", 0, 2, "MALE",$$
$$"sinan", "PROPER", "UNKNOWN", "PERSON", "SINGULAR").$$
$$mention("architect", 65, 18, "ANIMATE", 0, 1, "MALE",$$
$$"architect", "PROPER", "UNKNOWN", "O", "SINGULAR").$$
$$roleappositive(64, 65).$$

$$attribute\_agreement\_for\_stage(42).$$
$$link(ID1, ID2, 43) \leftarrow roleappositive(ID1, ID2), firstmenofclu(ID2, 42, C2),$$
$$ID1 < ID2, notprune(ID2, 42, C2), notsamecluster(ID1, ID2, 42),$$
$$attributeagreement(C1, C2, "number", 42),$$
$$attributeagreement(C1, C2, "animacy", 42),$$
$$attributeagreement(C1, C2, "gender", 42),$$
$$attributeagreement(C1, C2, "ner", 42),$$
$$attributeagreement(C1, C2, "person", 42), incluster(ID1, 42, C1).$$

4. Relative pronoun: the mention is a relative pronoun, and it modifies the head of the antecedent NP [16]. For example, *"[the finance street [which] has already formed in the Waitan district]..."*

$$mention("the\ new\ museum\ which\ has\ already\ opened\ in\ Istanbul$$
$$city", 71, 20, "INANIMATE", 0, 11, "MALE",$$
$$"museum", "NOMINAL", "UNKNOWN", "O", "SINGULAR").$$
$$mention("which", 72, 20, "INANIMATE", 3, 4, "MALE", "museum",$$
$$"PRONOMINAL", "UNKNOWN", "O", "SINGULAR").$$
$$relativepronoun(71, 72).$$

Rules

$$link(ID1, ID2, 44) \leftarrow relativepronoun(ID1, ID2), firstmenofclu(ID2, 43, C2),$$
$$ID1 < ID2, notprune(ID2, 43, C2), notsamecluster(ID1, ID2, 43).$$

5. Acronym: One of the mentions is an acronym of the other, and both are taged as *NNP* [16]. For example, *United Nations* and *UN* .

$$link(ID1, ID2, 45) \leftarrow acronym(ID1, ID2), firstmenofclu(ID2, 44, C2),$$
$$ID1 < ID2, notprune(ID2, 44, C2), notsamecluster(ID1, ID2, 44).$$

6. Demonym: one of the mentions is a demonym of the other. For demonym detection Wikipedia static list of countries and their gentilic have been used [16], for example, *Syria*, and *Syrian*.

$$link(ID1, ID2, 46) \leftarrow demonym(ID1, ID2), firstmenofclu(ID2, 45, C2),$$
$$ID1 < ID2, notprune(ID2, 45, C2), notsamecluster(ID1, ID2, 45).$$

Model 5: Strict Head Match

Linking two mentions if they have the same head word, and ignoring the possibility that they might have incompatible modifiers can generate many wrong links. To overcome this issue, many conditions must be all considered to link mentions with the same head word [19].

1. Entity head match: the head word of the mention to be considered matches the head word of any mention in the antecedent entity (cluster) [16]. For example

$$mention(''architect,\ mimar\ sinan'', 7, 1, ''ANIMATE'', 0, 4, ''MALE'',$$
$$''architect'', ''NOMINAL'', ''UNKNOWN'', ''O'', ''SINGULAR'').$$
$$mention(''mimar\ sinan'', 4, 1, ''ANIMATE'', 2, 4, ''MALE'', ''sinan'',$$
$$''PROPER'', ''UNKNOWN'', ''PERSON'', ''SINGULAR'').$$
$$mention(''sinan'', 11, 2, ''ANIMATE'', 0, 1, ''MALE'', ''sinan'',$$
$$''PROPER'', ''UNKNOWN'', ''PERSON'', ''SINGULAR'').$$
$$incluster(11, 4, 10).$$
$$incluster(7, 4, 3).$$
$$incluster(4, 4, 3).$$

Head match will be detected between the mentions

$$headmatch(11, 7, 51).$$

$$headmatch(11, 4, 51).$$

And the head match link

$$headmatch(ID1, ID2, 51) \leftarrow ID1 < ID2, firstmenofclu(ID2, 4, C2), notprune(ID2, 4, C2), hwmen(ID2, HW), hwsc(C1, HW, 4), incluster(ID1, 4, C1), nominal(ID1), nominal(ID2).$$

2. Word inclusion: the non-stop word set of the current entity (mention's cluster) is a subset of the non-stop words set of the antecedent entity [16]. The list of non-stop words that have been considered in Stanford

$$(”a”, ”an”, ”the”, ”of”, ”at”, ”on”, ”upon”, ”in”, ”to”, ”from”, ”out”, ”as”, ”so”, ”such”, ”or”, ”and”, ”those”, ”this”, ”these”, ”that”, ”for”, ”,”, ”is”, ”was”, ”am”, ”are”, ”'s”, ”been”, ”were”, ”mr.”, ”miss”, ”mrs.”, ”dr.”, ”ms.”, ”inc.”, ”ltd.”, ”corp.”)$$

For example the mentions are linked

$$mention(”president\ recep\ tayyip\ erdogan”, 4, 1, ”ANIMATE”, 0, 4, ”MALE”, ”erdogan”, ”PROPER”, ”UNKNOWN”, ”PERSON”, ”SINGULAR”).$$

$$incluster(4, 4, 3).$$

$$word(4, ”president”, 0, ”nonstop”, 7).$$

$$word(4, ”recep”, 1, ”nonstop”, 8).$$

$$word(4, ”tayyip”, 2, ”nonstop”, 9).$$

$$word(4, ”erdogan”, 3, ”nonstop”, 10).$$

$$mention(”the\ president”, 7, 2, ”ANIMATE”, 0, 2, ”MALE”, ”president”, ”NOMINAL”, ”UNKNOWN”, ”O”, ”SINGULAR”).$$

$$incluster(7, 4, 5).$$

$$word(7, ”the”, 0, stop”, 14).$$

$$word(7, ”president”, 1, ”nonstop”, 15).$$

As the word inclusion are considered in several passes, this rule is used to activate it for those passes

$$word\_inclusion\_for\_stage(P)$$

Nonstop words *W* of the cluster *C* in pass *P*

$$nonswofclu\_(W, C, P) \leftarrow word\_inclusion\_for\_stage(P),$$
$$incluster(ID, P, C), word(ID, W, WS, "nonstop", WSS).$$

Number of nonstop words of a cluster *C* in pass *P*

$$nonswocc(C, X, P) \leftarrow word\_inclusion\_for\_stage(P), cluster(C, P),$$
$$X = \#count\{W, C, P : nonswofclu\_(W, C, P)\}.$$

For pairs of distinct clusters *C1*, *C2* and nonstop word *W* that occur in both clusters in pass *P*

$$nonsw\_has\_partner(C1, C2, W, P) \leftarrow word\_inclusion\_for\_stage(P),$$
$$nonswofclu\_(W, C1, P), nonswofclu\_(W, C2, P),$$
$$cluster(C1, P), cluster(C2, P), C1 != C2.$$

Nonstop word has no partner i.e. word *W* is in cluster *C1* but not in cluster *C2* in pass *P*

$$nonsw\_has\_no\_partner\_w(C1, C2, W, P) \leftarrow word\_inclusion\_for\_stage(P),$$
$$nonswofclu\_(W, C1, P), C1 != C2,$$
$$notnonsw\_has\_partner(C1, C2, W, P),$$
$$cluster(C1, P), cluster(C2, P).$$

If there is a nonstop word that is in *C1* but not in *C2*

$$nonsw\_has\_no\_partner(C1, C2, P) \leftarrow word\_inclusion\_for\_stage(P),$$
$$nonsw\_has\_no\_partner\_w(C1, C2, \_, P),$$
$$cluster(C1, P), cluster(C2, P), C1 != C2.$$

Cluster of the mention must have at least one nonstop word

$$cluster\_has\_nonsw(C, P) \leftarrow word\_inclusion\_for\_stage(P),$$
$$nonswocc(C, X, P), X > 0.$$

Activation of word inclusion for clusters in 4th pass, and linking the mentions

$$word\_inclusion\_for\_stage(4).$$
$$linkwordinclusion(ID1, ID2, 52) firstmenofclu(ID2, 4, C2), ID1 < ID2,$$
$$not\ nonsw\_has\_no\_partner(C2, C1, 4),$$
$$cluster\_has\_nonsw(C2, 4), incluster(ID1, 4, C1).$$

3. Compatible modifiers only: This condition focuses on the two individual mentions not the entities, i.e. the mention's modifier set is included in the modifier set of the other mention. Compatible modifiers will be satisfied if the modifiers of any mention in the entity of the mention to be considered are contained in the modifiers of any mention in the antecedent candidate's entity.

$mention("several\ major\ cities\ in\ the\ last\ week", 10, 1, "INANIMATE",$

$12, 19, "UNKNOWN", "cities", "NOMINAL", "UNKNOWN", "O", "PLURAL").$

$mod(10, 1, "the", "DT").$

$mod(10, 2, "last", "JJ").$

$mod(10, 3, "week", "NN").$

$mod(10, 4, "several", "JJ").$

$mod(10, 5, "major", "JJ").$

$mention("the\ last\ week", 11, 1, "INANIMATE", 16, 19, "UNKNOWN",$

$"week", "NOMINAL", "UNKNOWN", "O", "SINGULAR").$

$mod(11, 1, "the", "DT").$

$mod(11, 2, "last", "JJ").$

The atom *linkmod(ID1,ID2)* represents the compatible modifier relation between mentions *ID1*, and *ID2*.

Noun and adjective modifiers of the mention

$$\begin{aligned} nounadjmod(ID, S) \leftarrow\ & mod(ID, S, MM, T), T == ("NN"; "NNS"; "NNP"; \\ & "JJ"; "JJR"; "JJS"; "CD"; "VB"; "VBD"; \\ & "NNPS"; "VBG"; "VBN"; "VBP"; "VBZ"). \end{aligned}$$

Number of the modifiers (adjectives and nouns) of each mention

$$\begin{aligned} modomenc(ID2, X) \leftarrow\ & X = \#count\{ID2, S2, MM2, T2 : mod(ID2, S2, MM2, T2), \\ & nounadjmod(ID2, S2)\} > 0, mention(M2, ID2, \\ & SN2, ANIM2, SI2, EI2, G2, HW2, MT2, P2, NER2, ISPR2). \end{aligned}$$

The modifier $MM2$ of the mention $ID2$ to be solved is found in the modifier set of $ID1$

$foundmod(ID1, MM2, S2, T2, ID2, 53) \leftarrow mod(ID1, S1, MM1, T1),$

$$mod(ID2, S2, MM2, T2),$$
$$nounadjmod(ID1, S1), nounadjmod(ID2, S2),$$
$$MM1 == MM2, ID1 < ID2,$$
$$not\ samecluster(ID1, ID2, 4).$$

Number of modifiers of the mention *ID2* that are found in the modifier set of *ID1*

$$\begin{aligned} exactmodc(ID1, ID2, C, 53) \leftarrow\ & C = \#count\{ID1, MM2, S2, T2, ID2, 53 : \\ & foundmod(ID1, MM2, S2, T2, ID2, 53)\} > 0, \\ & notsamecluster(ID1, ID2, 4), \\ & mention(M1, ID1, SN1, ANIM1, SI1, EI1, G1, HW1, \\ & MT1, P1, NER1, ISPR1), mention(M2, ID2, SN2, \\ & ANIM2, SI2, EI2, G2, HW2, MT2, P2, NER2, ISPR2). \end{aligned}$$

All the modifiers of the mention *ID2* are found in the modifier set of *ID1*, the mentions are linked

$$\begin{aligned} linkmod(ID1, ID2) \leftarrow\ & exactmodc(ID1, ID2, Y), modomenc(ID2, Y), \\ & nominal(ID1), nominal(ID2). \end{aligned}$$

Compatible modifiers only for a pass will be satisfied if we have *linkmod(IDX,IDY)* between at least two mentions *IDY* form the mention's cluster, and *IDX* from the antecedent candidate's cluster, four cases have been considered

- *linkmod(IDX,IDY)* between the mention, and its antecedent candidate

$$\begin{aligned} linkmodfinal(ID1, ID2, 53) \leftarrow\ & linkmod(ID1, ID2), nominal(ID1), \\ & nominal(ID2), firstmenofclu(ID2, 4, C2), \\ & not\ prune(ID2, 4, C2), ID1 < ID2, \\ & notsamecluster(ID1, ID2, 4), headmatch(ID1, ID2, 51). \end{aligned}$$

- *linkmod(IDX,IDY)* between the mention, and one mention in the antecedent candidate's cluster other than the antecedent candidate.

$$\begin{aligned} linkmodfinal(ID1, ID2, 53) \leftarrow\ & linkmod(ID3, ID2), samecluster(ID1, ID3, 4), \\ & notsamecluster(ID1, ID2, 4), ID3 != ID1, \end{aligned}$$

$$nominal(ID1), nominal(ID2),$$
$$firstmenofclu(ID2, 4, C2),$$
$$notprune(ID2, 4, C2), ID1 < ID2,$$
$$headmatch(ID1, ID2, 51).$$

- *linkmod(IDX,IDY)* between a mention form the cluster of the mention to be solved, and its antecedent candidate

$$\begin{aligned} linkmodfinal(ID1, ID2, 53) \leftarrow\ & linkmod(ID1, ID4), \\ & samecluster(ID2, ID4, 4), \\ & notsamecluster(ID1, ID2, 4), ID4! = ID2, \\ & nominal(ID1), nominal(ID2), \\ & firstmenofclu(ID2, 4, C2), \\ & notprune(ID2, 4, C2), ID1 < ID2, \\ & headmatch(ID1, ID2, 51). \end{aligned}$$

- *linkmod(IDX,IDY)* between two mentions from the clusters of the mention to be solved, and its antecedent candidate other than those two mentions

$$\begin{aligned} linkmodfinal(ID1, ID2, 53) \leftarrow\ & linkmod(ID3, ID4), samecluster(ID2, ID4, 4), \\ & samecluster(ID1, ID3, 4), nominal(ID1), \\ & not\ samecluster(ID1, ID2, 4), nominal(ID2), \\ & ID4! = ID2, ID3! = ID1, \\ & firstmenofclu(ID2, 4, C2), notprune(ID2, 4, C2), \\ & ID1 < ID2, headmatch(ID1, ID2, 51). \end{aligned}$$

4. Not i-within-i : i-within-i construct means one mention is a child NP in the other's NP constituent [16].

$$mention(\text{"}the\ spirit\ of\ the\ people\text{"}, 6, 2, \text{"}INANIMATE\text{"}, 6, 11, \text{"}NEUTRAL\text{"},$$
$$\text{"}spirit\text{"}, \text{"}NOMINAL\text{"}, \text{"}UNKNOWN\text{"}, \text{"}O\text{"}, \text{"}SINGULAR\text{"}).$$
$$mention(\text{"}the\ people\text{"}, 7, 2, \text{"}ANIMATE\text{"}, 9, 11, \text{"}UNKNOWN\text{"}, \text{"}people\text{"},$$
$$\text{"}NOMINAL\text{"}, \text{"}UNKNOWN\text{"}, \text{"}O\text{"}, \text{"}PLURAL\text{"}).$$

If one mention is a child NP in the others NP constituent from Stanford we got the fact

$$includedin(ID1, ID2).$$

$$includedin(6, 7).$$

Mentions *ID1*, and *ID2* must not have appositive, relative pronoun, or role appositive (mentioned in model 4) relation.

$$\begin{aligned} norelation(ID1, ID2) \leftarrow\ & not\ appositive(ID1, ID2), not\ appositive(ID2, ID1), \\ & not\ relativepronoun(ID1, ID2), not\ relativepronoun(ID2, ID1), \\ & notroleappositive(ID1, ID2), notroleappositive(ID2, ID1), \\ & mention(M1, ID1, SN1, ANIM1, SI1, EI1, G1, \\ & HW1, MT1, P1, NER1, ISPR1), mention(M2, ID2, SN2, \\ & ANIM2, SI2, EI2, G2, HW2, MT2, P2, NER2, ISPR2), \\ & ID1 != ID2. \end{aligned}$$

And the mentions have i-within-i

$$\begin{aligned} linkiwi(ID1, ID2, 54) \leftarrow\ & firstmenofclu(ID2, 4, C2), ID1 < ID2, \\ & norelation(ID1, ID2), headmatch(ID1, ID2, 51), \\ & includedin(ID1, ID2), nominal(ID1), nominal(ID2), \\ & not\ prune(ID2, 4, C2), not\ samecluster(ID1, ID2, 4). \end{aligned}$$

If the above four constraints are satisfied, the mention and its antecedent candidate are linked

$$\begin{aligned} link(ID1, ID2, 5) \leftarrow\ & headmatch(ID1, ID2, 51), linkwordinclusion(ID1, ID2, 52), \\ & linkmodfinal(ID1, ID2, 53), notlinkiwif(ID1, ID2, 54). \end{aligned}$$

Models 6 and 7: Variants of Strict HeadMatch

These models are different relaxations of the conditions in model 5, where in model 6 there is no compatible modifiers only condition [16] i.e.

$$\begin{aligned} link(ID1, ID2, 6) \leftarrow\ & headmatch(ID1, ID2, 61), \\ & linkwordinclusion(ID1, ID2, 62), not\ linkiwif(ID1, ID2, 63). \end{aligned}$$

And model 7 removes the word inclusion constraint[16] i.e.

$$link(ID1, ID2, 7) \leftarrow headmatch(ID1, ID2, 71), linkmodfinal(ID1, ID2, 72),$$

$$notlinkiwif(ID1, ID2, 73).$$

Model 8: Proper Head Word Match

Mentions that their head words are proper nouns are linked if they have the same head word and satisfy the following conditions [16]:

- Are not in i-within-i
- No location mismatches: all the location named entities, proper nouns of the mention to be considered are included in those of the antecedent candidate
- No numeric mismatches : all the number of the mention to be considered are included in those of the antecedent candidate.

Mention *ID* with a proper noun head word

$$properheadwordmen(ID) \leftarrow headwordtype(ID, T), T == ("nnp"; "nnps").$$

Mentions with the same proper noun head word

$$\begin{aligned} sameproperheadwordmens(ID1, ID2) \leftarrow\ & hwmen(ID2, HW), hwmen(ID1, HW), \\ & properheadwordmen(ID1), properheadwordmen(ID2), \\ & not\ samecluster(ID1, ID2, 7), ID1 < ID2, \\ & firstmenofclu(ID2, 7, C2), not\ prune(ID2, 7, C2). \end{aligned}$$

For the no location mismatches condition between mentions, location mismatch can happen in two cases

1. If the head word of one mention is *"country"* or *"nation"*, and the other mention text is a name of state, or an abbreviation of state name, the atom *statemencountrymen(ID1,ID2)* are used to indicate that between mentions. Stanford have been used to get facts to indicate if the mention is state or abbreviation of state name

$$stateorabb(ID).$$

$$\begin{aligned} statemencountrymen(ID1, ID2) \leftarrow\ & stateorabb(ID1), hwmen(ID2, HW), \\ & HW == ("country"; "nation"). \end{aligned}$$

2. Location modifiers of one mention are not included in the other mention location modifiers, the strategy that have been used in compatible modifier only (in model 5) have been used here, i.e. matching location modifier and counting the matches between mentions. Location modifiers of a mention *ID*

$$\begin{aligned} loactionmod(ID, S) \leftarrow\ & mod(ID, S, MM, POS), \\ & loctagofmod(ID, S, LOCT), LOCT == "LOCATION". \end{aligned}$$

Counting of mention's location modifiers

$$\begin{aligned} locmodomenc(ID2, X) \leftarrow\ & X = \#count\{ID2, S2, MM2, T2 : \\ & mod(ID2, S2, MM2, T2), loactionmod(ID2, S2)\}, \\ & mention(M2, ID2, SN2, ANIM2, SI2, EI2, G2, \\ & HW2, MT2, P2, NER2, ISPR2). \end{aligned}$$

Location modifier of the mention has a match in the other mention's modifiers

$$\begin{aligned} foundlocmod(ID1, MM2, S2, T2, ID2) \leftarrow\ & mod(ID1, S1, MM1, T1), mod(ID2, S2, MM2, T2), \\ & loactionmod(ID1, S1), loactionmod(ID2, S2), \\ & MM1 == MM2, notsamecluster(ID1, ID2, 7), \\ & ID1 != ID2. \end{aligned}$$

Counting location modifier matches between mentions

$$\begin{aligned} exactlocmodc(ID1, ID2, C) \leftarrow\ & C = \#count\{ID1, MM2, S2, T2, ID2 : \\ & foundlocmod(ID1, MM2, S2, T2, ID2)\}, \\ & mention(M1, ID1, SN1, ANIM1, SI1, EI1, G1, \\ & HW1, MT1, P1, NER1, ISPR1), \\ & mention(M2, ID2, SN2, ANIM2, SI2, EI2, G2, \\ & HW2, MT2, P2, NER2, ISPR2), ID1 != ID2. \end{aligned}$$

Location modifiers of one mention are included in location modifier of the other mention

$$\begin{aligned} linkmodlocation(ID1, ID2) \leftarrow\ & exactlocmodc(ID1, ID2, X), locmodomenc(ID2, X), \\ & nominal(ID1), nominal(ID2), not\ prune(ID2, 7, C2), \\ & not\ statemencountrymen(ID1, ID2), , ID1 < ID2 \end{aligned}$$

$$sameproperheadwordmens(ID1, ID2),$$
$$not\ samecluster(ID1, ID2, 7), firstmenofclu(ID2, 7, C2).$$

No proper nouns modifier mismatches: the same strategy as in location mismatch.

proper noun modifiers

$$propermod(ID, S) \leftarrow mod(ID, S, MM, T), T == ("NNP"; "NNPS").$$

Counting the proper noun modifiers

$$propermodomenc(ID2, X) \leftarrow X = \#count\{ID2, S2, MM2, T2 :$$
$$mod(ID2, S2, MM2, T2), propermod(ID2, S2)\},$$
$$mention(M2, ID2, SN2, ANIM2, SI2, EI2, G2, HW2,$$
$$MT2, P2, NER2, ISPR2).$$

Matching proper noun modifiers

$$foundpropermod(ID1, MM, S2, T2, ID2) \leftarrow mod(ID1, S1, MM, T1),$$
$$mod(ID2, S2, MM, T2), propermod(ID1, S1),$$
$$propermod(ID2, S2), ID1 != ID2,$$
$$notsamecluster(ID1, ID2, 7).$$

Counting proper noun modifier matches between mentions

$$exactpropermodc(ID1, ID2, C) \leftarrow C = \#count\{ID1, MM2, S2, T2, ID2 :$$
$$foundpropermod(ID1, MM2, S2, T2, ID2)\},$$
$$mention(M1, ID1, SN1, ANIM1, SI1, EI1, G1,$$
$$HW1, MT1, P1, NER1, ISPR1).$$

Proper noun modifiers of one mention are included in the other mention location modifier

$$linkmodproperlocation(ID1, ID2) \leftarrow xactpropermodc(ID1, ID2, Y),$$
$$propermodomenc(ID2, X), Y == X, nominal(ID1),$$
$$nominal(ID2), linkmodlocation(ID1, ID2),$$
$$firstmenofclu(ID2, 7, C2), notprune(ID2, 7, C2),$$
$$ID1 < ID2, not\ samecluster(ID1, ID2, 7).$$

No numeric mismatches: same as location modifiers and noun proper modifiers. Mentions have to be in the same sentence, with word distance less than six .

$$\begin{aligned} linkmensdistance(ID1, ID2) \leftarrow\, & -mention(M1, ID1, SN1, ANIM1, SI1, \\ & EI1, G1, HW1, MT1, P1, NER1, ISPR1), \\ & mention(M2, ID2, SN2, ANIM2, SI2, EI2, G2, \\ & HW2, MT2, P2, NER2, ISPR2), SN1 == SN2, \\ & |SI2 - SI2| < 6, firstmenofclu(ID2, 7, C2), ID1 < ID2, \\ & not\ prune(ID2, 7, C2), not\ samecluster(ID1, ID2, 7), \\ & sameproperheadwordmens(ID1, ID2). \end{aligned}$$

Mentions are linked if all the previous conditions are satisfied with attribute agreement between clusters of the mentions

$attribute\_agreement\_for\_stage(7).$

$$\begin{aligned} link(ID1, ID2, 8) \leftarrow\, & linkmensdistance(ID1, ID2), linkmodnumber(ID1, ID2), \\ & linkmodproperlocation(ID1, ID2), not\ linkiwi(ID1, ID2, 81), \\ & attributeagreement(C1, C2, ''number'', 7), \\ & incluster(ID1, 7, C1), \\ & attributeagreement(C1, C2, ''animacy'', 7), \\ & attributeagreement(C1, C2, ''gender'', 7), \\ & attributeagreement(C1, C2, ''ner'', 7), \\ & attributeagreement(C1, C2, ''person'', 7). \end{aligned}$$

Model 9: Relaxed Head Match

Both mentions are labeled as named entities from the same type, and the head word of the mention matches any word in the antecedent entity. Also this model implements a combination of those conditions with not i-within-i and word Inclusion [16].

Head word of the mention *hwmen(ID2,HW)* matches a word *wofclu(HW,WS,C1,8)* in the antecedent candidate's cluster

$$\begin{aligned} headwmatchw(ID1, ID2) \leftarrow\, & hwmen(ID2, HW), wofclu(HW, WS, C1, 8), \\ & not\ samecluster(ID1, ID2, 8), ID1 < ID2, \\ & firstmenofclu(ID2, 8, C2), incluster(ID1, 8, C1), \end{aligned}$$

$$notprune(ID2, 8, C2).$$

Mentions are labeled as names entities, and have the same type

$$\begin{aligned} samenamedentitymens(ID1, ID2) \leftarrow\ & mention(M1, ID1, SN1, ANIM1, SI1, EI1, G1, \\ & HW1, MT1, P1, NER1, ISPR1), mention(M2, ID2, \\ & SN2, ANIM2, SI2, EI2, G2, HW2, MT2, P2, \\ & NER2, ISPR2), NER1 != "O", NER2 != "O", \\ & not\ samecluster(ID1, ID2, 8), \\ & not\ prune(ID2, 8, C2), ID1 < ID2, NER1 == NER2, \\ & firstmenofclu(ID2, 8, C2), headwmatchw(ID1, ID2). \end{aligned}$$

Mentions are linked if the mentioned conditions are all satisfied

$$\begin{aligned} link(ID1, ID2, 9) \leftarrow\ & samenamedentitymens(ID1, ID2), \\ & linkwordinclusion(ID1, ID2, 92), not\ linkiwi(ID1, ID2, 91). \end{aligned}$$

Model 10: Pronominal Coreference

All the previously mentioned models do coreference between nominal mentions except for model 1, model 4. But that does not mean that the Sieve system ignores pronominal coreference. The mentioned nine models make the system ready for this model through building precise clusters with shared mention attributes, which is so important for pronominal coreference [16], Distance between the two mentions must be at most 3 sentences.
Sentence of the mention

$$mensen(ID, SN) \leftarrow mention(M, ID, SN, ANIM, SI, EI, G, HW, T, P, NER, ISPR).$$

Activate the attribute agreement for model 9

$$attribute\_agreement\_for\_stage(9).$$

Link pronouns with nominal antecedent candidate, if there are agreements in every attribute between them

$$\begin{aligned} link(ID1, ID2, 10) \leftarrow\ & ID1 < ID2, notsamecluster(ID1, ID2, 9), \\ & firstmenofclu(ID2, 9, C2), pronominal(ID2), nominal(ID1), \end{aligned}$$

$$\begin{aligned}
&attributeagreement(C1, C2, \text{"}number\text{"}, 9),\\
&attributeagreement(C1, C2, \text{"}animacy\text{"}, 9),\\
&attributeagreement(C1, C2, \text{"}gender\text{"}, 9),\\
&attributeagreement(C1, C2, \text{"}ner\text{"}, 9),\\
&attributeagreement(C1, C2, \text{"}person\text{"}, 9),\\
&incluster(ID1, 9, C1), mensen(ID1, SN1), mensen(ID2, SN2),\\
&SN2 - SN1 < 4, incluster(ID2, 9, C2).
\end{aligned}$$

Link pronouns with pronominal antecedent candidate, if there is an agreement in all attribute except for the person attribute

$$\begin{aligned}
link(ID1, ID2, 10) \leftarrow\ & ID1 < ID2, notsamecluster(ID1, ID2, 9),\\
& firstmenofclu(ID2, 9, C2), pronominal(ID2), pronominal(ID1),\\
& attributeagreement(C1, C2, \text{"}number\text{"}, 9),\\
& attributeagreement(C1, C2, \text{"}animacy\text{"}, 9),\\
& attributeagreement(C1, C2, \text{"}gender\text{"}, 9),\\
& attributeagreement(C1, C2, \text{"}ner\text{"}, 9),\\
& incluster(ID1, 9, C1), mensen(ID1, SN1),\\
& mensen(ID2, SN2), SN2 - SN1 < 4, incluster(ID2, 9, C2).
\end{aligned}$$

Cluster of the mentions that are linked according to a model will be merged, merge is an equivalence relation, i.e. it is reflexive, transitive, and symmetric.

$$\begin{aligned}
mergecluster(C1, C2, SN) \leftarrow\ & nextStage(S, SN), incluster(MI1, S, C1),\\
& incluster(MI2, S, C2), link(MI1, MI2, S), C1 < C2.\\
mergecluster(C1, C2, S) \leftarrow\ & mergecluster(C2, C1, S), activatePass(S).\\
mergecluster(C1, C1, S) \leftarrow\ & mergecluster(C1, _, S), activatePass(S).\\
mergecluster(C1, C3, S) \leftarrow\ & mergecluster(C1, C2, S), mergecluster(C2, C3, S),\\
& C1 < C2, C2 < C3, activatePass(S).
\end{aligned}$$

We identify the lexicographically smallest element of the equivalence relation, and we define atoms of form ***mergeInto(Ct,Cs)*** for source clusters ***Cs*** that are merged into target clusters ***Ct***.

$$mergeBelow(C2) \leftarrow mergeCluster(C1, C2), C1 < C2.$$

$$mergeInto(Ct, Cs) \leftarrow mergeCluster(Ct, Cs), not\ mergeBelow(Ct).$$

Clusters that will cease to exist in model S.

$$\begin{aligned} abandonCluster(S, COld) \leftarrow\ & activatePass(S), CNew < COld, \\ & mergeInto(CNew, COld, S). \end{aligned}$$

Clusters that have been extended i.e. merged with other clusters

$$\begin{aligned} incluster(MI, SN, CNew) \leftarrow\ & incluster(MI, S, COld), \\ & mergeInto(CNew, COld, SN), \\ & activatePass(SN), nextStage(S, SN). \end{aligned}$$

Unchanged clusters in model S.

$$\begin{aligned} incluster(MI, SN, COld) \leftarrow\ & activatePass(SN), nextStage(S, SN), \\ & notabandonCluster(SN, COld), \\ & incluster(MI, S, COld). \end{aligned}$$

Chapter 3

Results and Discussion

3.1 CORPORA

We used OntoNotes-Dev – development partition of OntoNotes v 5.0 [28] corpora for development and formal evaluation. The OntoNotes project [28] is a collaborative annotation effort conducted by BBN Technologies and several universities, the goal was to annotate large corpus from various genres with syntax, propositional structure, named entities and word senses, as well as coreference resolution.

The OntoNotes coreference annotation view files (.coref files) are our input files, those files are formatted using in-line annotation. XML tag called COREF, and were used to indicate the beginning and the end of the mentions that should be linked, with ID attribute indicating the ID of the chain that contains the linked mentions see figure 3.1 . Coreference view follows the Treebank tokenization, and also includes the trace and empty category elements ("*", "*-2", and "*-1") found in the Treebank analysis, since those can also participate in the coreference chains.

3.2 EVALUATION

We used four evaluation metrics B^3 (Bagga and Baldwin 1998), $CEAF$ (Constrained Entity Aligned F-measure), MUC (Vilain et al. 1995), $BLANC$ (BiLateral Assessment of NounPhrase Coreference),and $CoNLLF1$ to compare our ASP Sieve with the Stanford Sieve. The official CoNLL scorer v8.01 (CoNLL2011 shared task Pradhan et al. 2011) have been used, it takes as input CoNLL

```
<DOC DOCNO="bc/cnn/00/cnn_0003@0003@cnn@bc@en@on">

<TEXT PARTNO="000">

..

..

<COREF ID="m_5" TYPE="IDENT" SPEAKER="Linda_Hamilton">I</COREF> mean
<COREF ID="m_5" TYPE="IDENT" SPEAKER="Linda_Hamilton">I</COREF> have *-
1 to tell you that when <COREF ID="m_5" TYPE="IDENT"
SPEAKER="Linda_Hamilton">I</COREF> married <COREF ID="122"
TYPE="IDENT"><COREF ID="123" TYPE="APPOS" SUBTYPE="ATTRIB"><COREF
ID="m_5" TYPE="IDENT" SPEAKER="Linda_Hamilton">my</COREF> first
husband</COREF> <COREF ID="123" TYPE="APPOS"
SUBTYPE="HEAD">Bruce</COREF></COREF> *T*-2 <COREF ID="m_5" TYPE="IDENT"
SPEAKER="Linda_Hamilton">I</COREF> <COREF ID="m_5" TYPE="IDENT"
SPEAKER="Linda_Hamilton">I</COREF> <COREF ID="138"
TYPE="IDENT">went</COREF> into hiding for the first year /.

<COREF ID="m_5" TYPE="IDENT" SPEAKER="Linda_Hamilton">I</COREF> just
started *-1 reading books /.

<COREF ID="m_5" TYPE="IDENT" SPEAKER="Linda_Hamilton">I</COREF> mean
<COREF ID="122" TYPE="IDENT">the man</COREF> married <COREF ID="m_5"
TYPE="IDENT" SPEAKER="Linda_Hamilton">me</COREF> /.

and <COREF ID="m_5" TYPE="IDENT" SPEAKER="Linda_Hamilton">I</COREF>
<COREF ID="138" TYPE="IDENT">went</COREF> into hiding for a year /.

..

..

<COREF ID="m_26" TYPE="IDENT" E_OFF="1" SPEAKER="Linda_Hamilton">I-
</COREF> <COREF ID="m_26" TYPE="IDENT"
SPEAKER="Linda_Hamilton">I</COREF> 'm sure 0 there is *?* /.

Um if <COREF ID="m_26" TYPE="IDENT" SPEAKER="Linda_Hamilton">I</COREF>
were <COREF ID="m_14" TYPE="IDENT" SPEAKER="caller_7">you</COREF> ,
because <COREF ID="m_26" TYPE="IDENT"
SPEAKER="Linda_Hamilton">I</COREF> do n't know <COREF ID="43"
TYPE="IDENT">that number</COREF> off hand um <COREF ID="m_14"
TYPE="IDENT" SPEAKER="caller_7">you</COREF> can call <COREF ID="70"
TYPE="IDENT">the University of Medicine and Dentistry in <COREF ID="50"
TYPE="IDENT">New Jersey</COREF></COREF> /.

Um oh <COREF ID="74" TYPE="IDENT">they</COREF> would have *-1 to know
where in <COREF ID="50" TYPE="IDENT">New Jersey</COREF> then /.

..

..

</TEXT>

</DOC>
```


Figure 3.1: OntoNotes coreference annotation view

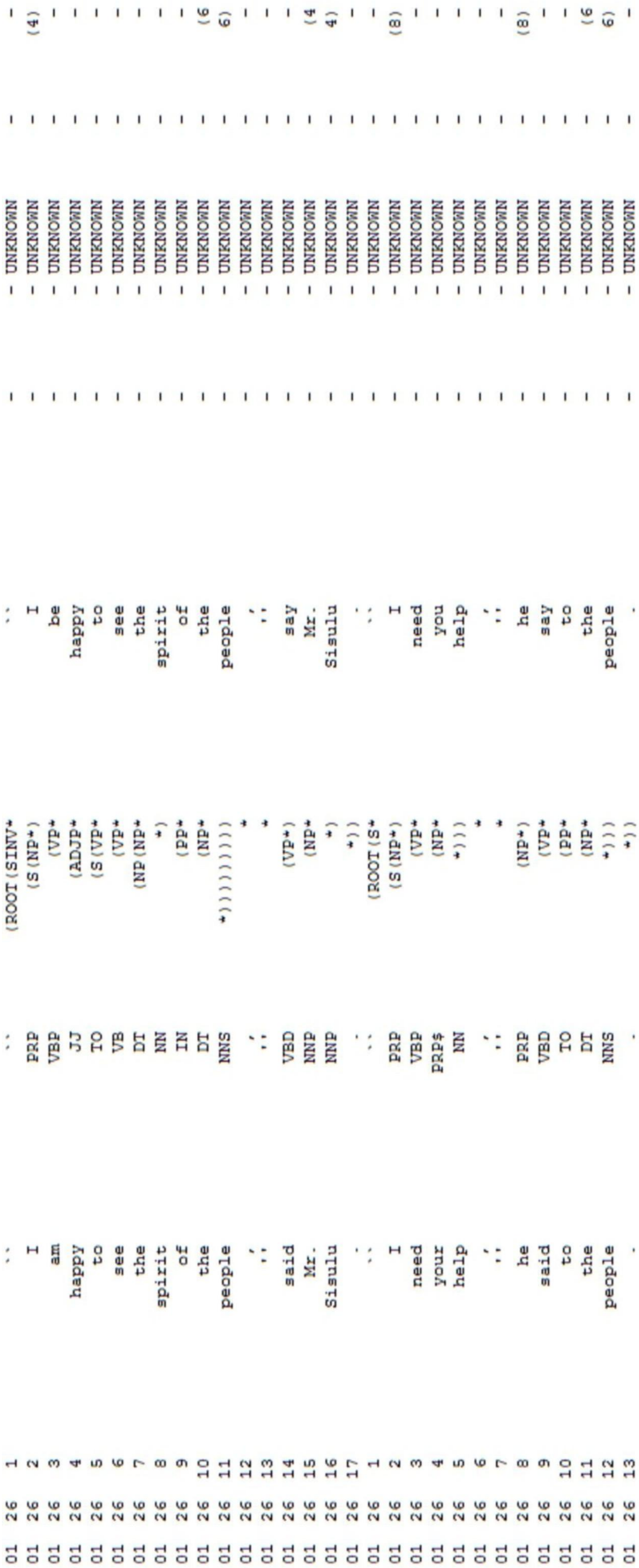
files. In general CoNLL file contains a representation of all the OntoNotes layers in the CoNLL style tabular format one line per token, where the last column represents coreference information. The columns in the CoNLL file are [23] figure 3.2

1. Document ID.
2. Part number: Some documents have several texts.
3. Word number: word index in the sentence.
4. The word.
5. Word's part of speech.
6. Parse bit: Bracketed structure broken before the first open parenthesis in the parse, and the word/part-of-speech leaf replaced with a *.
7. Predicate lemma: is mentioned if we have the semantic role information of the row if not the rows are marked with a -.
8. Predicate Frameset ID: The PropBank frameset ID of the predicate in Column 7.
9. Word sense.
10. Author/Speaker.
11. Named Entities.
12. Predicate Arguments: There is one column each of predicate argument structure information for the predicate mentioned in Column 7.
13. Coreference: Coreference chain ID.

The scorer does not need all the columns to work, it needs just four columns (document ID, part number, word itself, coreference).

3.3 ASP SIEVE IMPLEMENTATION

Figure 3.3 shows how our system works, as input we have OntoNotes coreference annotation view files (.coref files), from which several things will be extracted



```
01 26  1      ``   ``   (ROOT(SINV*      ``   -  -  UNKNOWN  -  -
01 26  2       I   PRP  (S(NP*)           I   -  -  UNKNOWN  -  (4)
01 26  3      am   VBP  (VP*             be   -  -  UNKNOWN  -  -
01 26  4   happy   JJ   (ADJP*        happy   -  -  UNKNOWN  -  -
01 26  5      to   TO   (S(VP*           to   -  -  UNKNOWN  -  -
01 26  6     see   VB   (VP*            see   -  -  UNKNOWN  -  -
01 26  7     the   DT   (NP(NP*         the   -  -  UNKNOWN  -  -
01 26  8  spirit   NN   *)           spirit   -  -  UNKNOWN  -  -
01 26  9      of   IN   (PP*             of   -  -  UNKNOWN  -  -
01 26 10     the   DT   (NP*            the   -  -  UNKNOWN  -  (6
01 26 11  people   NNS  *)))))))     people   -  -  UNKNOWN  -  6)
01 26 12       ,   ,    *                 ,   -  -  UNKNOWN  -  -
01 26 13      ''   ''   *                ''   -  -  UNKNOWN  -  -
01 26 14    said   VBD  (VP*)           say   -  -  UNKNOWN  -  -
01 26 15     Mr.   NNP  (NP*            Mr.   -  -  UNKNOWN  -  (4
01 26 16  Sisulu   NNP  *)           Sisulu   -  -  UNKNOWN  -  4)
01 26 17       .   .    *))               .   -  -  UNKNOWN  -  -
01 26  1      ``   ``   (ROOT(S*         ``   -  -  UNKNOWN  -  -
01 26  2       I   PRP  (S(NP*)           I   -  -  UNKNOWN  -  (8)
01 26  3    need   VBP  (VP*           need   -  -  UNKNOWN  -  -
01 26  4    your   PRP$ (NP*            you   -  -  UNKNOWN  -  -
01 26  5    help   NN   *)))           help   -  -  UNKNOWN  -  -
01 26  6       ,   ,    *                 ,   -  -  UNKNOWN  -  -
01 26  7      ''   ''   *                ''   -  -  UNKNOWN  -  -
01 26  8      he   PRP  (NP*)            he   -  -  UNKNOWN  -  (8)
01 26  9    said   VBD  (VP*            say   -  -  UNKNOWN  -  -
01 26 10      to   TO   (PP*             to   -  -  UNKNOWN  -  -
01 26 11     the   DT   (NP*            the   -  -  UNKNOWN  -  (6
01 26 12  people   NNS  *)))         people   -  -  UNKNOWN  -  6)
01 26 13       .   .    *))               .   -  -  UNKNOWN  -  -
```

Figure 3.2: CoNLL file format

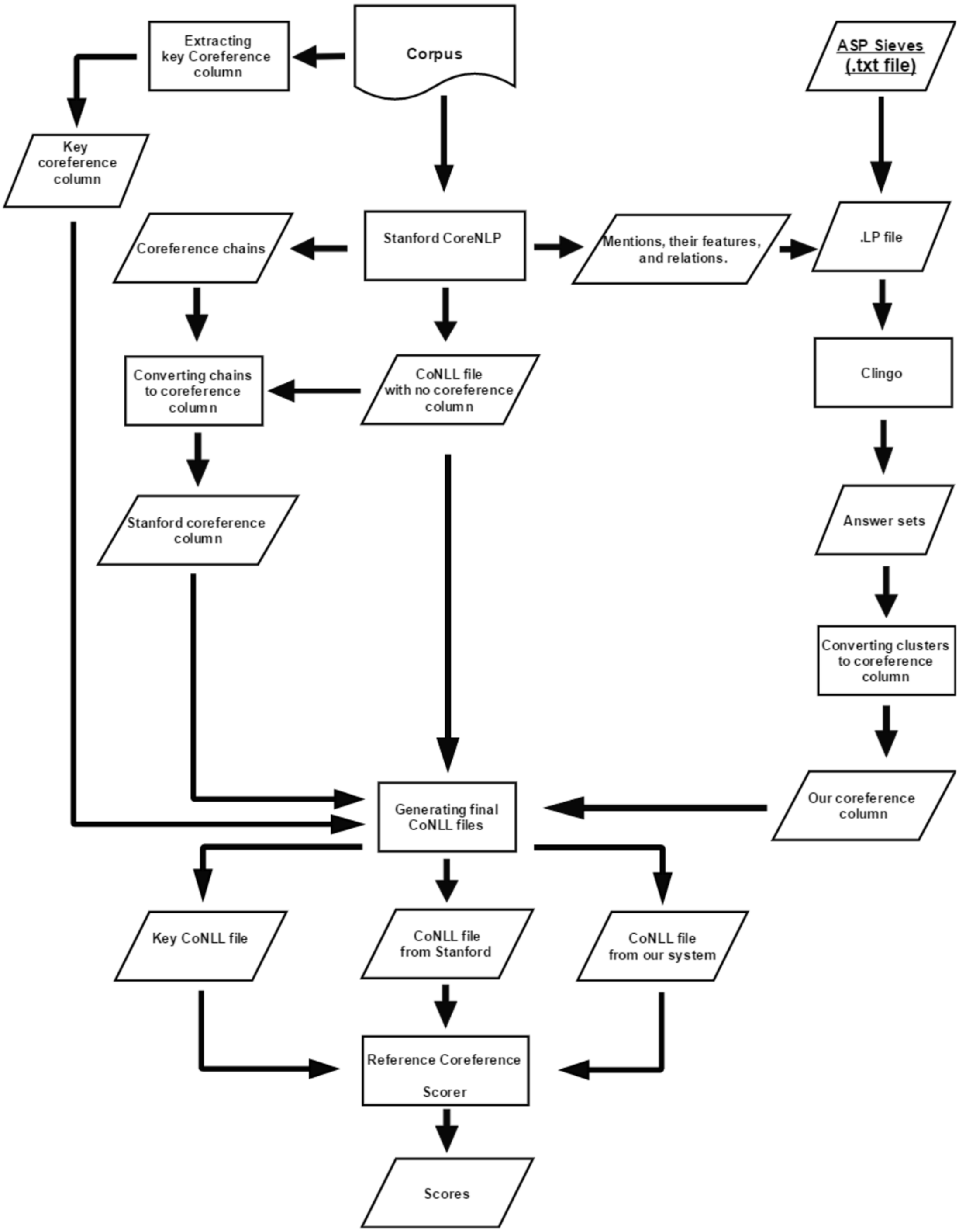


Figure 3.3: Our system

1. CoNLL file with no coreference column.

2. Mentions, their attributes, and their relations as .txt file. This file will be combined with our ASP implementation of the Sieve, generating .LP file, this file will be given to the ASP Solver. The Solver gives us as result .txt file. The file indicates in which cluster each mention ends up with according to our ASP Sieve.

3. Coreference chains(clusters) form Stanford CoreNLP, these clusters indicates in which cluster each mention ends up with according to Stanford CoreNLP.

Stanford CoreNLP tools contain a MUC format reader however this reader does not work on OntoNotes data, neither does it work on original MUC6 or MUC7 corpora since an internal XML API changed. As a result we had to implement three java tools to convert the results to CoNLL format in order to compare them

1. Our ASP Sieve result to CoNLL file.

2. Stanford Sieve result to CoNLL file.

3. OntoNotes coreference annotation view(. coref files) to CoNLL file, this is the key file that will be compared with the result files from our ASP Sieve and Stanford Sieve.

The scorer takes as input the three CoNLL files to give us the final scores for both our ASP Sieve, and Stanford Sieve.

3.4 EXPERIMENTAL RESULTS

Our current prototype system performs coreference resolution with a logic program, and an ASP solver. We have implemented all passes of the original Stanford CoreNLP deterministic coreference resolution Sieve [16] in Answer Set Programming and made experiments on OntoNotes-Dev – development partition [28]. Tables 3.1 show the the results from Stanford Sieve and our ASP Sieve, regarding correctness of our approach we have not yet adjusted the ASP rules so that they fully capture the logic realized in the Stanford Java code. Currently we obtain a performance which is 5% below the MUC-score of Stanford this is expected as we here just show the feasibility of doing the Sieve fully declarative in ASP.

Models	MUC			B^3			CEAF			BLANC		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1
Stanford Model(1)	11.82	87.08	20.81	6.22	67.59	11.39	1.23	35.03	2.37	9.24	59.26	14.63
ASP Sieve Model(1)	3.5	88.05	6.73	1.31	86.19	2.58	1.23	35.03	2.37	1.94	78.82	3.77
Stanford+(2,3)	29.05	69.76	41.02	19	61.48	29.03	20.44	42.13	27.53	17.31	55.84	26.05
ASP Sieve+(2,3)	18.93	47.51	27.07	13.67	46.18	21.1	21.65	31.26	25.58	8.99	46.29	14.34
Stanford+(4,5,6,7)	35.09	56.26	41.22	26.4	48.54	34.2	31.18	39.33	34.79	22.26	46.82	30.17
ASP Sieve+(4,5,6,7)	23.59	41.73	30.14	19.28	39.57	25.92	27.93	32.81	30.18	11.2	36.4	16.64
Stanford+(8,9)	35.58	56.1	43.54	26.9	48.3	34.56	31.69	39.7	35.23	22.54	46.68	30.4
ASP Sieve+(8,9)	24.96	41.24	31.1	20.19	38.4	26.46	28.09	33.01	30.35	12.54	34.91	18.05
Stanford+(10)	57.89	55.8	56.83	45.73	47.23	46.47	46.72	41.11	43.74	47.08	47.67	46.92
ASP Sieve+(10)	53.2	50.21	51.66	37.31	42.68	39.82	37.83	33.41	35.48	40.28	39.45	39.46

Table 3.1: Experimental results

Figure 3.4 shows the distribution of time required to process documents in this experimental run: from 355 documents, the majority was processed in less than 1 second; a few documents required between 10 and 35 seconds which is slower than the Java implementation of the Sieve.

3.5 Conclusion

So far we encountered several challenges in this project and learned several important lessons:

- Rule-based Coreference Resolution uses a complex set of rules, that is hard to realize based on scientific literature only, yet the available source code improves the situation.
- Answer Set Programming is a framework that allows for realizing all these rules in a natural way, however for efficiency reasons we need encoding techniques that make the logic program appear less natural;
- Performance in the deterministic case is reasonable, apart from few very big documents which require around 30 seconds to process;
- Each tool and each corpus has its own data format, and converters rarely work without tweaking them manually.

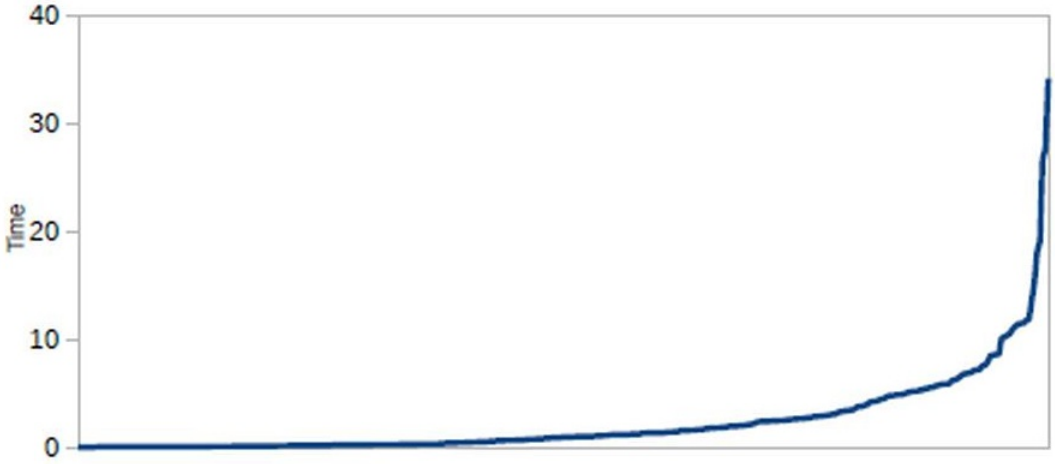


Figure 3.4: Required times (in seconds) for ASP reasoning in TEST portion of the English OntoNotes corpus, sorted by the required time.

In the following we briefly discuss each of these points and provide an outlook on possible future research directions.

Although the paper describing Stanford Sieve [16] is clearly written and very explicit, in several cases important information has been omitted or formulated in an ambiguous way, such that it was necessary to read the Java source code to improve correspondence of our ASP implementation with Stanford. In fact, so far we were not able to fully reproduce the CoreNLP result on coreference in our ASP Sieve, because we have not yet looked at the complete source code. We use the same input that dcoref gets in ASP, so theoretically a 100% equal implementation should be possible. An example, the constraint *Compatible modifiers only* which have been used in several models has the following definition *"the mention's modifiers are all included in the modifiers of the antecedent candidate...."* In our first implementation of this constraint we applied it to the mention to be solved, and its antecedent candidate, but we have found in the Java code was so different, If the modifier of at least one mention from the cluster of the mention to be solved are included in the modifier set of at least one mention in the antecedent candidate's cluster this constraint will be satisfied . Another example is attribute agreement between cluster, according to [16], it can be understood that this constraints have been used just in Pronominal Coreference Resolution (tenth model), but we found that it has been considered in several models (fourth, eighth). We conclude that reproducibility is improved by providing source code, but it would be better if the scientific description is precise enough to allow a nearly exact reimplementation.

ASP is a very different style of programming compared to procedural languages like Java. Our initial representation for the Sieve modules was beautiful but its performance was bad. That is mainly because a naive implementation of Sieve modules and its preconditions in ASP rules will represent all conditions for all layers, regardless of whether a layer requires such a condition or not. ASP is usually applied to combinatorial optimization problems, where programs are small but very hard to solve, different from what we do: large programs that are easy to solve. Therefore ASP does not provide automatic optimization mechanisms that eliminate unused representations from the program. Moreover debugging support for ASP is still in its infancy and also difficult to apply if answer sets need post processing in Java to visualize the result. The Java Sieve implementation realizes rules using if statements and function calls, and some functions are only called if certain conditions are met. In ASP we represent truth of conditions, instantiate rules if certain conditions are potentially true, and this way obtain the result. This can be wasteful regarding the amount of represented conditions, and appropriate encoding techniques can ameliorate this problem at the cost of making the logic program less intuitive to read and harder to maintain. We chose ASP for using a formalism with nondeterminism that mirrors the ambiguity of natural language, and because ASP provides constructions for connecting rules with knowledge, in particular semantic web knowledge, see, e.g. [10]. To use these strengths of ASP, however, we first need to overcome more basic issues like optimizing encodings to make them feasible for applications with large programs. We conclude that an important future direction for such applications will be to automatically optimize (non-ground) ASP programs, a task that recently gained increased attention in the community [6].

We found a standard coreference scorer software which uses the widely used CoNLL format, however the OntoNotes corpus which is stored in MUC format does not contain a converter into CoNLL format (it contains a Python API and other Python tools). Moreover the Stanford CoreNLP tools contain a MUC format reader however this reader does not work on OntoNotes data. As a result we implemented our own MUC-to-CoNLL converter.

3.6 Summary

Stanfords Sieve [16] is a deterministic approach to coreference resolution that combines the global information and precise features of modern machine-learning models with the modularity of deterministic, rule-based systems. In this work, we implemented the Stanford Sieve architecture using computational logic. We represent the Sieve and its modules completely in the rule-based ASP [9, 12, 18] formalism, which is a general purpose declarative logic programming formalism that supports comfortable representation of knowledge, nonmonotonic reasoning processes, and reasoning with hybrid knowledge bases, and made experiments on the OntoNotes 5.0 corpus. Regarding correctness of our approach we have not yet adjusted the ASP rules so that they fully match the logic of Stanford Sieve Java code. Currently we obtain a performance which is 5% below the MUC-score of Stanford Sieve.

Bibliography

[1] Alviano, M., Dodaro, C., Faber, W., Leone, N., Ricca, F.: Wasp: A native asp solver based on constraint learning. In: International Conference on Logic Programming and Nonmonotonic Reasoning. pp. 54–66. Springer (2013)

[2] Baral, C., Dzifcak, J.: Solving puzzles described in english by automated translation to answer set programming and learning how to do that translation. arXiv preprint arXiv:1108.3850 (2011)

[3] Baral, C., Gelfond, G., Son, T.C., Pontelli, E.: Using answer set programming to model multi-agent scenarios involving agents' knowledge about other's knowledge. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1. pp. 259–266. International Foundation for Autonomous Agents and Multiagent Systems (2010)

[4] Brewka, G., Eiter, T., Truszczyński, M.: Answer set programming at a glance. Communications of the ACM 54(12), 92–103 (2011)

[5] Brooks, D.R., Erdem, E., Erdoğan, S.T., Minett, J.W., Ringe, D.: Inferring phylogenetic trees using answer set programming. Journal of Automated Reasoning 39(4), 471–511 (2007)

[6] Buddenhagen, M., Lierler, Y.: Performance tuning in answer set programming. In: International Conference on Logic Programming and Nonmonotonic Reasoning. pp. 186–198. Springer (2015)

[7] Delgrande, J.P., Grote, T., Hunter, A.: A general approach to the verification of cryptographic protocols using answer set programming. In: International Conference on Logic Programming and Nonmonotonic Reasoning. pp. 355–367. Springer (2009)

[8] Egly, U., Alice Gaggl, S., Woltran, S.: Answer-set programming encodings for argumentation frameworks. Argument and Computation 1(2), 147–177 (2010)

[9] Eiter, T., Ianni, G., Krennwallner, T.: Answer set programming: A primer. In: Reasoning Web. Semantic Technologies for Information Systems, pp. 40–110. Springer (2009)

[10] Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. Artificial Intelligence 172(12), 1495–1539 (2008)

[11] Erdem, E., Yeniterzi, R.: Transforming controlled natural language biomedical queries into answer set programs. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing. pp. 117–124. Association for Computational Linguistics (2009)

[12] Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Answer set solving in practice. Synthesis Lectures on Artificial Intelligence and Machine Learning 6(3), 1–238 (2012)

[13] Gebser, M., Kaufmann, B., Kaminski, R., Ostrowski, M., Schaub, T., Schneider, M.: Potassco: The potsdam answer set solving collection. Ai Communications 24(2), 107–124 (2011)

[14] Hirst, G.: Anaphora in natural language understanding (1981)

[15] Law, M., Russo, A., Broda, K.: Simplified reduct for choice rules in asp. Tech. rep., Tech. Rep. DTR2015-2, Imperial College of Science, Technology and Medicine, Department of Computing (2015)

[16] Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., Jurafsky, D.: Deterministic coreference resolution based on entity-centric, precision-ranked rules. Computational Linguistics 39(4), 885–916 (2013)

[17] Lierler, Y., Schüller, P.: Parsing combinatory categorial grammar via planning in answer set programming. In: Correct Reasoning, pp. 436–453. Springer (2012)

[18] Lifschitz, V.: Answer set programming and plan generation. Artificial Intelligence 138(1), 39–54 (2002)

[19] Lifschitz, V.: What is answer set programming?. In: AAAI. vol. 8, pp. 1594–1597 (2008)

[20] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: ACL (System Demonstrations). pp. 55–60 (2014)

[21] Mitkov, R.: Anaphora resolution: the state of the art. Citeseer (1999)

[22] Peng, H., Khashabi, D., Roth, D.: Solving hard coreference problems. Urbana 51, 61801 (2015)

[23] Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., Xue, N.: Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task. pp. 1–27. Association for Computational Linguistics (2011)

[24] Pührer, J.: Stepwise Debugging in Answer-Set Programming. Ph.D. thesis, Technische Universität Wien (2014)

[25] Ricca, F., Grasso, G., Alviano, M., Manna, M., Lio, V., Iiritano, S., Leone, N.: Team-building with answer set programming in the gioia-tauro seaport. Theory and Practice of Logic Programming 12(03), 361–381 (2012)

[26] Tang, C.K.F., Ternovska, E.: Model checking abstract state machines with answer set programming. Fundamenta Informaticae 77(1-2), 105–141 (2007)

[27] Tiihonen, J., Soininen, T., Niemelä, I., Sulonen, R., et al.: A practical tool for mass-customising configurable products. In: DS 31: Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm (2003)

[28] Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., Franchini, M., et al.: Ontonotes release 4.0. LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium (2011)

[29] Zheng, J., Chapman, W.W., Crowley, R.S., Savova, G.K.: Coreference resolution: A review of general methodologies and applications in the clinical domain. Journal of biomedical informatics 44(6), 1113–1122 (2011)

Resume

Kenda Alakraa

PERSONAL INFORMATION

Nationality: Syrian.

Mobile: 00905358387176.   ***Birth:*** 6/2/ 1984

Email: kinda_akraa @yahoo.com

OBJECTIVE

To be associated with an enterprise, where my educational and technical capabilities will lead to a responsible position and have mutual benefits.

EDUCATION

Qualification:

- Bachelor degree in Informatics Engineering [72,495]
  - Specialized in Software Engineering. Syria, Albaath Univ. [2006]
- Master degree in computer engineering [91]
  - Specialized in Natural Language Processing . Turkey, Marmara Univ. [2016]

Courses:

Operating Systems, Parallel Programming, Software Engineering, Information Security, Distributed Systems, Database, Data Communication & Computer Networks, Compilers, Quality & Reliability, Information Technology, Algorithms & Data Structure, and Project Management.

EXPERIENCE

- Worked as Technical support & web site administrator at Health Science College, Albaath Univ. One semester [Syria 2009]

- I have been working in some special institute in Homs as teacher

  Professional international center (2005-2012)
  Almamoun (2007-2009)

- Taught Software engineering for 5th year students of Informatics Engineering

  College, Albaath Univ. One semester [Syria 2007].

- Taught Parallel programming for 4th year students of Informatics Engineering

  College, Albaath Univ. One semester [Syria 2007].

- Taught E commerce for 5th year students of Informatics Engineering College, Albaath Univ. One semester [Syria 2007].

- Taught Software engineering 2 for 4th year students of Informatics Engineering

  College, Albaath Univ. One semester [Syria 2009].

- Conducted many seminars and lectures at the university.

- I have been working as a teacher in Computer Technical Institute at Al-Baath

  Univ . [Syria 2009-2012].

TECHNICAL SKILLS

Programming languages:

.NET, C#,Java.

PHP.

Operating systems:

MS Windows 2000, XP.

PROJECTS

- ***English learning.*** ***2004***

  Educational CD oriented to primary school students to help them to learn English. [Used techs: Macromedia Authorware, Flash and Photoshop].

- ***Encryption & decryption software.*** ***2005***

Software encrypts and decrypts files or texts. [Used techs: C# .net]

- ***Mobile messenger.*** **2005**

  SMS chatting program on Symbian OS. [Used techs: J2ME].

- ***Graduation project*** ***: COSES (****Component based OS for Embedded Systems****)*** **2006**

  Graphical software with code generator to help OS programmers design OS for Embedded systems aided by components, to reduce time and effort wasting and make the OS as much as reliable and errorless. [Used techs: NesC Language, C#.Net ,Linux environment, Tiny OS Components].

- ***Master project*** ***:*** *Coreference Resolution Sieve Based on Answer Set Programming[TUBITK]* ***2017***

  Coreference Resolution is the task of connecting phrases and prepositions in a text if they denote the same real world entity. Our system performs coreference resolution with a logic program, and an ASP solver.

LANGUAGES

- ***Arabic:*** native language.
- ***English:*** Reading (Excellent), Writing (Excellent),Speaking (Very good)

  TOFEL Certificate(Paper Based) :563 , March 2009

  WELT Certificate :Grammar and Usage C ,Writing B, Reading B, May 2009

  YDS Certificate: 60, April 2013

- ***Turkish:*** Reading (very good),Writing (very good),Speaking (Very good)

  C1 Certificate: 77, June 2013