Multi-Sieve Pass for Coreference Resolution

Summarized by Rodrigo Agerri
IXA NLP Group
UPV/EHU
rodrigo.agerri@ehu.es

Abstract

This document introduces the multi-sieve pass for coreference resolution system developed by the NLP Group of Stanford University (Raghunathan et al., 2010; Lee et al., 2011) and (Lee et al., 2013). This system proposes a number of deterministic passes, ranging from high precision to higher recall, each dealing with a different manner in which coreference manifests itself in running text.

Contents

1 Introduction **3**

2 Mention Processing **3**
2.1 Mention Detection . . . 4
2.2 Resolution Architecture . . . 5
2.2.1 Filtering the Search Space: Mention Selection in a Given Sieve . . . 5
2.2.2 Antecedent Selection for a Given Mention . . . 6
2.2.3 Feature Sharing . . . 6

3 Sieve Passes **7**
3.1 Exact Match . . . 8
3.2 Precise Constructs . . . 8
3.3 Strict Head Match . . . 9
3.4 Pronominal Coreference Resolution . . . 10

4 Running Example **11**
4.1 Mention Detection . . . 11
4.2 Speaker Detection Sieve . . . 11
4.3 Exact String Match Sieve . . . 12
4.4 Precise Constructs . . . 13
4.5 Strict Head Match . . . 13
4.6 Pronoun Match . . . 13

5 Datasets and required NLP tools **14**
5.1 NLP tools . . . 14
5.2 Datasets . . . 15

6 Roadmap **15**

7 Annex A: Detecting Pleonastic 'It' **17**

List of Tables

1 Multi-sieve Pass and CoNLL 2011 F1 Evaluation . . . 4
2 Running Coreference Resolution Example. . . . 12
3 Coreference Roadmap. . . . 16

List of Figures

1 Left-to-right breadth first tree traversal. . . . 7
2 i-within-i construct. . . . 9

1 Introduction

This document describes the procedure required to develop a coreference resolution system based on the Stanford's system (Lee et al., 2013) and (Raghunathan et al., 2010; Lee et al., 2011). In principle the various versions of the system presented in three different publications consist of several sieve passes, which can be summarized in the 10 passes listed in table 1 (Lee et al., 2013).

Each sieve pass is applied in a deterministic manner, reusing the information generated by the previous sieve and the mention processing. The order in which the sieves are a applied favours a highest precision approach and aims at improving the recall with the subsequent application of each of the sieve passes.

This is illustrated by the evaluation results of the CoNLL 2011 Coreference Evaluation task (Lee et al., 2011, 2013), listed in the right hand column of table 1, in which the Stanford's system obtained the best results. The results show a pattern which has also been shown in other results reported with other evaluation sets (Raghunathan et al., 2010), namely, the fact that a large part of the performance of the multi pass sieve system is based on few of the sieves. Thus, the results show that sieves 1, 2, 5 and 10 provide 97% of the results for that particular evaluation set (Lee et al., 2011, 2013).

This is **GOOD NEWS!** then, as partners responsible for each language could prioritize their work **to focus on a subset of sieves only** (plus the mention detection and processing of course), namely, Exact Match, Precise Constructs, Strict Head Match and Pronoun Match. However, it should be clarified that some sieves not introduced in this document, notably sieve 1 *Speaker Match* are important for subsequent sieves. However, for the first coreference resolution system, implementing 4 sieves for the 6 OpeNER languages will be a considerable achievement in itself. Furthermore, the Speaker Match sieve added unwanted complexity at this stage of development. Finally, looking at domain specific datasets should clarify if this sieve is going to have any real impact on the overall results of the system.

In the following sections, we describe each step. First the detection and processing of mentions in section 2. Section 3 describes each of the 4 aforementioned sieves: Exact String Match (3.1), Strict Head Match (3.3) and Pronominal Coreference Resolution (3.4). We then provide a running example in section 4 which would help to understand the flow of information between the sieves and the mention processing modules. In the last two sections, we list the NLP requirements for the development of the coreference resolution module (section 5) and we establish a ROADMAP with specific steps to be taken (including a tentative calendar) for the development of such a module in each of the OpeNER languages (section 6).

2 Mention Processing

The Mention processing module consists of the mention detection and the mention resolution architecture:

Sieves	Type	CONLL 2011 F1
Mention Detection	NPs, NER and PRP	-
Sieve 1	**Speaker Identification**	**29.2**
Sieve 2	**Exact String Match**	**45.3**
Sieve 3	Relaxed String Match	45.4
Sieve 4	Precise Constructs	45.7
Sieve 5	**Strict Head Match A**	**48.5**
Sieve 6	Strict Head Match B	48.8
Sieve 7	Strict Head Match C	49.3
Sieve 8	Proper Head Noun Match	49.5
Sieve 9	Relaxed Head Match	49.7
Sieve 10	**Pronoun Match**	**59.3**

Table 1: Multi-sieve Pass and CoNLL 2011 F1 Evaluation

- *Mention Detection*: Noun Phrases (NP), Named Entity Recognition (NERC), Pronouns (PRP) and heuristics.
- *Resolution Architecture*:
    1. **Reduce search space**: consider only first mentions of each cluster to be resolved in the application of each sieve.
    2. **Selection of antecedents for each sieve**: left to right breadth first search.
    3. **Feature sharing**: gender, number and animacy, union for these attributes for every mention. If contradiction keep variety of attributes so that later the cluster can be linked with any mention which has any of these attributes.

2.1 Mention Detection

Their detection algorithm is based on syntactic features, NER and manually-written patterns. The manually written patterns are highly corpus-dependent, so even though we will describe them because they are used in the English coreference module, it should be noted that these patterns will probably change according to the annotation policy of each corpus. For example, the adjectival forms of nations are valid in the ACE corpus (LDC et al., 2005).

1. **Start by marking all noun phrases, pronouns and named entity mentions** (no previously marked as modifiers in the NPs).
2. **Apply the following heuristics or patterns:**
    (a) Remove a mention if a larger mention with the same head word exists. For example, remove *The five insurance companies* in *The five insurance companies approved to be established this time.*

(b) Discard numeric entities such as percents, money, cardinals, and quantities: 9%, $10, *Tens of thousands*, *100 miles*.

(c) Remove mentions with partitive or quantification: *a total of 177 projects*, *none of them*, *millions of people*. As a general rule, these are the NPs with 'of' preceded by one of 9 quantifiers and 34 partitives in English.

(d) Remove pleonastic *it* pronouns, detecting using patterns such as *It is possible that ...*, *It seems that ...*, *It turns out ...*. They list the patterns specified in Annex A 7.

(e) Discard adjectival forms of nations or nationality acronyms: *American*, *U.S.*, following Ontonotes 4.0 annotation guidelines (Weischedel et al., 2010).

(f) Remove these words: *there, ltd.,etc,'s, and hmm.*

Again, it should be stressed that **patterns are corpus-dependent**. Therefore, if we are to design these type of patterns, it should be considered that:

1. **For the first year of OpeNER**, they will need to be created for each different evaluation corpus for each language of OpeNER.

2. **For the second year**, these patterns will need to be created for each language for the evaluation datasets of reviews that need to be created.

2.2 Resolution Architecture

The resolution architecture incrementally creates mention clusters in each sieve pass application. As mentioned above, the resolution architecture at each sieve pass **(i) it filters the search space for which mention should be considered for resolution** (see section 2.2.1) , **(ii) it sorts the antecedents that should be considered for resolution of a given mention** (see section 2.2.2), and **(iii) it constructs features from partially-built mention clusters** (section 2.2.3).

2.2.1 Filtering the Search Space: Mention Selection in a Given Sieve

For each sieve pass, we have partial mention clusters produced by the previous sieve pass. Of course, the first sieve pass (Exact Match) simply starts with singleton mention clusters, one for each of the mentions detected in the mention detection module (section 2.1). This information is used by:

1. Considering **only mentions** that are **first in textual order** in their cluster. Thus, if we are given the following ordered list of mentions: $m_1^1, m_2^2, m_3^2, m_4^3, m_5^1, m_6^2$, the coreference system will only try to resolve mention m_2^2 and mention m_4^3. These are the first mentions in textual order in the mention clusters to which they belong[1]. It is easy to check that

[1]The *superscript* indicates *mention cluster id* and the *subscript* indicates the mention id

mention m_1^1 does not need to be resolved because it is the first in the text, and that the rest of the mentions are not the first in mention clusters 2 and 1.

2. **Discourse Salience**: Mentions appearing *first* in their mention clusters *are not to be considered for resolution* if:

   (a) The mention starts with **indefinite pronouns** such as *some, other*, etc.

   (b) The mention starts with **indefinite articles** such as *a, an*, etc.

   (c) The mentions are **bare plurals**, such as *bars, restaurants*, etc.

EXCEPTION!!: *Discourse Salience* does **NOT APPLY** for the *Exact String Match* Sieve (section 3.1).

2.2.2 Antecedent Selection for a Given Mention

Once we are given a given mention m_i, each sieve pass either will not be able to propose a solution depending on its features, or *will deterministically select a single best antecedent mention* from a list of previously ordered mentions $m_1, \ldots, m_i$. In Stanford's multi sieve pass system, the *candidate antecedents are ordered* using syntactic information from a *constituent parsing analysis*:

1. In given sentential clause, namely, those parser **constituents** which start with an **S label**, are sorted using a left-to-right breadth-first traversal of the corresponding syntactic constituent. Figure 1 shows an example of candidate ordering based on this method (Hobbs, 1978).

2. If the sentence containing the mention to be resolved contains multiple clauses, the above method is repeated separately for each $S*$ constituent, starting with the constituent in which the mention is located.

3. **Clauses in previous sentences** are sorted based on their textual proximity to the mention to be resolved.

The antecedent sorting is important because the coreference resolution system stops at the first match, namely, at the antecedent which has been placed as first by this selection method.

2.2.3 Feature Sharing

In the multi sieve pass system, each sieve pass gets (possibly incomplete) entity[2] information for each mention located in the mention clusters built by previous sieve passes. Mentions that has not been placed in any mention cluster or entity by any of the sieve passes are singleton members of their own mention cluster. The feature sharing procedure is as follows:

[2]Entity here refers to a cluster of mentions.

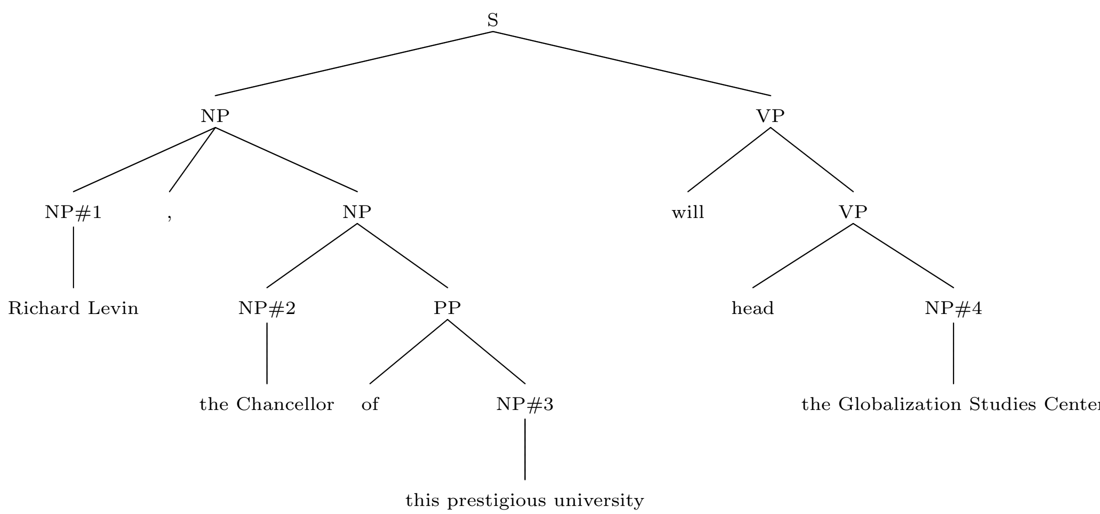

Figure 1: Left-to-right breadth first tree traversal.

1. **A union of every attribute for all mentions in a given mention cluster or entity is performed**. These attributes can be mention's number, gender, animacy, etc.

2. If attributes from two or more mentions in the same cluster contradict each other, then all attribute variants are maintained for that mention cluster or entity. For example, if two mentions in a given mention cluster display singular plural number respectively, then both attributes (singular/plural) are maintained as attributes of the mention cluster or entity.

3. Later, the mention cluster or entity displaying (singular/plural) attributes can be merged or accept further mentions that display both types of attributes.

3 Sieve Passes

As explained in the introduction and on table 1, we will be introducing only 4 types of sieve passes: Speaker Identification, Exact String Match, Strict Head Match and Pronoun Resolution. Previous evaluations on different corpora showed that by implementing these 4 sieves only we still get the large part of the coreference resolution performance. Therefore, this will reduce the amount of work and complexity needed to provide a complete coreference resolution system for each of the languages in OpeNER.

3.1 Exact Match

This model links two mentions only if they contain exactly the same extent text, including modifiers and determiners. For example, [the Shahab 3 ground-ground missile] and [the Shahab 3 ground-ground missile].

3.2 Precise Constructs

This pass links to mentions if any of the following conditions are satisfied:

1. **Appositive**: Two nominal mentions are in appositive construction, namely, "[Pierre Vinken], [chair of Elsevier], said...". The detection of apositives looks for the third children of a parent NP whose expansion begins with (NP, NP), when there is not a conjunction in the expansion (Haghighi and Klein, 2009).

2. **Predicate Nominative**: Two mentions, nominal or pronominal, are in copulative subject-object relation, for example, "[Pierre Vinken] is [the chairmain of Elsevier]" (Poon and Domingos, 2008).

3. **Role appositive**: the candidate antecedent is headed by a noun and appears as a modifier in a NP whose head is the current mention, for example, "[[actress] Rebecca Shaeffer]", inspired by Haghighi and Klein (2009), but further constrained as follows: This match will be considered if and only if:

   (a) The mention is labelled as a person.

   (b) The antecedent is animate (see 5.1).

   (c) The antecedent gender is not neutral.

4. **Relative Pronoun**: The mention is a relative pronoun that modifies the head of the antecedent NP: "[the finance street [which] has already formed in the Waitan district]".

5. **Acronym**: Both mentions are tagged as NNP and one of them is the acronym of the other: "[Agence France Presse], [AFP]".

6. **Demonym**: One of the mentions is a demonym of the other. For example, "[Mexico] and [Mexican]". This depends on a list of demonyms extracted from the Wikipedia.

Although in the evaluation table 1 this sieve does not provide a huge increase in performance, it plays a crucial role in providing precise information with respect to the mention attributes that is later used in the Pronouns Match sieve.

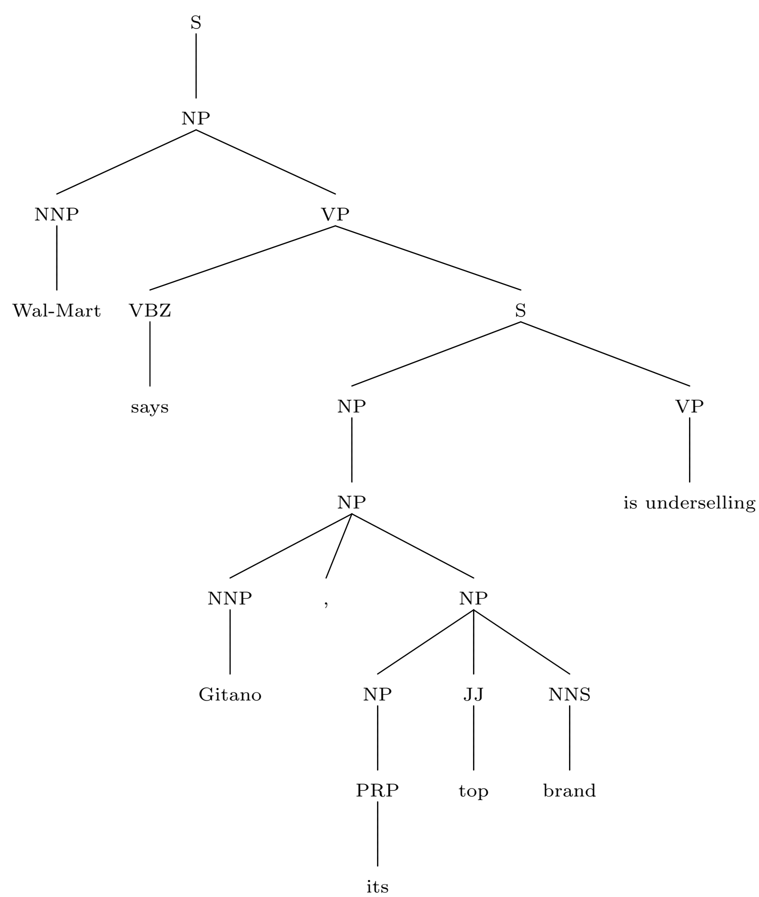

Figure 2: i-within-i construct.

3.3 Strict Head Match

Linking a mention to an antecedent based on the naive matching of their head words generates a lot of spurious links because it completely ignores possibly incompatible modifiers (Elsner and Charniak 2010). For example, *Yale University* and *Harvard University* have similar head words, but they are obviously different entities. To address this issue, this pass implements several constraints that **must all be matched** in order to yield a link:

1. **Entity head match**: The mention head word matches any head word of mentions in the antecedent entity. *This feature is constrained by enforcing a conjunction with the features below.*

2. **Word inclusion**: all the non-stop words in the current *entity* or mention cluster to be solved are included in the set of non-stop words in the *an-*

tecedent entity. For example, this pass correctly clusters together the two mentions in the following text:

(1) ... intervene in the [Florida Supreme Court]'s move ... does look like very dramatic change made by [the Florida court]

and avoids clustering the two mentions in the following text:

(2) The pilot had confirmed ... he had turned onto [the correct runway] but pilots behind him say he turned onto [the wrong runway].

3. **Compatible modifiers only**: the mention's modifiers, in this case only *nouns and adjectives*, are all included in the modifiers of the antecedent candidate. This feature models the same discourse property as the previous feature, but it focuses on the two individual mentions to be linked, rather than their corresponding entities or mention clusters.

4. **Not i-within-i**: the two mentions are not in an i-within-i construct. This means that one cannot be a child NP in the other's NP constituent. For example, in Figure 2 this prevents *its* from having the NP headed by *brand* in the set of possible antecedents. By propagation, it also removes the NP headed by *Gitano*. Therefore, it leaves the NP *Wal-Mart* as the closest compatible mention.

3.4 Pronominal Coreference Resolution

The implementation of this sieve pass is based on a well-known method, namely, enforcing constraints between the coreferent mentions. The following attributes are used for these constraints:

1. **Number**: Number attributes are assigned based on:
   (a) a static list for pronouns;
   (b) NER labels: mentions marked as a named entity are considered singular with the exception of organizations, which can be both singular and plural;
   (c) part of speech tags: NN*S tags are plural and all other NN* tags are singular;
   (d) a static dictionary from Bergsma and Lin (2006) [3].
2. **Gender**: gender attributes from static lexicons from Bergsma and Lin (2006) and from Ji and Lin (2009).
3. **Person**: Person attributes are assigned **only to pronouns**.
4. **Animacy**: Animacy attributes using:

[3] https://webdocs.cs.ualberta.ca/ bergsma/Gender/

(a) a static list for pronouns;

(b) NER labels, e.g., PERSON is animate whereas LOCATION is not;

(c) a dictionary bootstrapped from the Web (Ji and Lin, 2009).

5. **NER label**.

6. **Pronoun distance**: sentence distance between a pronoun and its antecedent cannot be larger than 3.

EXCEPTION!!: The *person constraint for pronouns* is **NOT APPLIED** when linking two pronouns if one appears within quotes. This is a simple heuristic for speaker detection, e.g., I and she point to the same person in "[I] voted my conscience," [she] said.

4 Running Example

In this section, we will explain using an example how the mention processing as described in section 2.1 and section 2.2 is performed at each of the sieves which will be applied sequentially (Lee et al., 2013). We will describe each row of table 2 separately to obtain a clear view of the overall flow of the multi sieve pass coreference resolution system.

4.1 Mention Detection

1. In mention detection (section 2.1) the system extracts mentions by detecting NPs and other modifier pronouns (PRP). We identify 11 different mentions assigned them initially to 11 different mention clusters or entities. Remember that *superscript* marks cluster or entity id whereas *subscript* marks mention id.

2. This step also extracts mention attributes from static lexicons referred to in section 3.4. For example, *John*{*ne:person*} and *A girl:*{*gender:female, number:singular*}.

The mention clusters which result after the application of the Mention Detection module form the input for the application of the sieve passes.

4.2 Speaker Detection Sieve

Although we do not provide an implementation of this sieve, we include this explanation for completeness reasons.

As explained in section 2.2.1, the selection of mentions (reducing search space) to be resolved is applied. In this case, as every mention cluster consists of one mention then every mention detected is a potential candidate for resolution (except the first mention in the text, obviously):

Input	John is a musician. He played a new song. A girl was listening to the song. “It is my favorite,” John said to her.
Mention Detection	[John]$^{1}_{1}$ is [a musician]$^{2}_{2}$. [He]$^{3}_{3}$ played [a new song]$^{4}_{4}$. [A girl]$^{5}_{5}$ was listening to [the song]$^{6}_{6}$. “[It]$^{7}_{7}$ is [[my]$^{9}_{9}$ favourite]$^{8}_{8}$,” [John]$^{10}_{10}$ said to [her]$^{11}_{11}$.
Speaker Sieve	[John]$^{1}_{1}$ is [a musician]$^{2}_{2}$. [He]$^{3}_{3}$ played [a new song]$^{4}_{4}$. [A girl]$^{5}_{5}$ was listening to [the song]$^{6}_{6}$. “[It]$^{7}_{7}$ is [[**my**]$^{9}_{9}$ favourite]$^{8}_{8}$,” [**John**]$^{9}_{10}$ said to [her]$^{11}_{11}$.
String Match	[**John**]$^{1}_{1}$ is [a musician]$^{2}_{2}$. [He]$^{3}_{3}$ played [a new song]$^{4}_{4}$. [A girl]$^{5}_{5}$ was listening to [the song]$^{6}_{6}$. “[It]$^{7}_{7}$ is [[my]$^{9}_{9}$ favourite]$^{8}_{8}$,” [**John**]$^{1}_{10}$ said to [her]$^{11}_{11}$.
Precise Constructs	[**John**]$^{1}_{1}$ is [**a musician**]$^{1}_{2}$. [He]$^{3}_{3}$ played [a new song]$^{4}_{4}$. [A girl]$^{5}_{5}$ was listening to [the song]$^{6}_{6}$. “[**It**]$^{7}_{7}$ is [[**my**]$^{1}_{9}$ **favourite**]$^{7}_{8}$,” [John]$^{1}_{10}$ said to [her]$^{11}_{11}$.
Strict Head Match	[John]$^{1}_{1}$ is [a musician]$^{1}_{2}$. [He]$^{3}_{3}$ played [**a new song**]$^{4}_{4}$. [A girl]$^{5}_{5}$ was listening to [**the song**]$^{4}_{6}$. “[It]$^{7}_{7}$ is [[my]$^{1}_{9}$ favourite]$^{7}_{8}$,” [John]$^{1}_{10}$ said to [her]$^{11}_{11}$.
Pronoun Match	[**John**]$^{1}_{1}$ is [a musician]$^{1}_{2}$. [**He**]$^{1}_{3}$ played [a new song]$^{4}_{4}$. [**A girl**]$^{5}_{5}$ was listening to [**the song**]$^{4}_{6}$. “[**It**]$^{4}_{7}$ is [[my]$^{1}_{9}$ favourite]$^{7}_{8}$,” [John]$^{1}_{10}$ said to [**her**]$^{5}_{11}$.

Table 2: Running Coreference Resolution Example.

1. **Mention Selection**: Traverse every mention in the text from left to right and choose the first mention in every cluster.
2. The first match for candidate to be resolved by the Speaker Detection sieve pass is mention my^9_9 (because of the quotes).
3. The sieve pass links my^9_9 with $John^{10}_{10}$ into the same mention cluster or entity (cluster id 9).

4.3 Exact String Match Sieve

This sieve looks for antecedents that have the exact same string as the mention under consideration.

1. **Mention Selection**: Note that *this does not apply* in this sieve pass. So we also obtain $John^9_{10}$ as a candidate mention even though is is not the

first mention in its cluster (my^9_9 is).

2. **Antecedent Selection**: As explained in section 2.2.2, candidate mentions are sorted from left to right using syntactic information. Thus, the resultant list of candidates for $John^9_{10}$ is: It^7_7, $My\ favorite^8_8$, My^9_9, $A\ girl^5_5$, $the\ song^6_6$, He^3_3, $a\ new\ song^4_4$, $John^1_1$, $a\ musician^2_2$.

3. **Exact String Match**: The algorithm stops when a matching antecedent is found. In this case, the algorithm will find $John^1_1$ and it stops there. Thus, $John^9_{10}$ changes mention cluster and becomes now $John^1_{10}$. The same goes to my^9_9: it now belongs to cluster 1 with $John^1_1$ and therefore becomes my^1_9.

4.4 Precise Constructs

The following row of table 2 shows the changes in the mention clusters when applying *Precise Constructs*. By applying this sieve pass $a\ musician^1_2$ is linked to $John^1_1$ and $my\ favorite^8_8$ is linked to It^7_7. In both cases the relevant construct is Subject-Predicate construction.

4.5 Strict Head Match

We do again the following steps:

1. **Mention Selection**: Textual order traversal of mentions to obtain first mention in their clusters as candidates. Among others, this provides $the\ song^6_6$ as candidate to be resolved.

2. **Antecedent Selection**: Left to right syntactic analysis procedure to obtain ordering of antecedents. This produces as the following order of antecedents for $the\ song^6_6$: $A\ girl^5_5$, He^3_3, $a\ new\ song^4_4$, and $John^1_1$.

3. **Strict Head Match**: Mentions without the same head word are removed. It only remains $a\ new\ song^4_4$.

4. **Strict Head Match**: non-stop words are all contained in antecedent and the 4 constraints of these sieve pass are held. Thus, $the\ song^6_6$ is becomes $the\ song^4_6$.

4.6 Pronoun Match

1. **Mention Selection**: We obtain as candidates for coreference linking He^3_3, It^7_7 and her^{11}_{11}.

2. **Antecedent Selection**: We obtain an antecedents ranking *for each of the candidates* obtained in the previous step. Thus, for He^3_3 we only get $John^1_1$ as antecedent. For It^7_7 we obtain $A\ girl^5_5$, He^3_3, $a\ new\ song^4_4$ and $John^1_1$.

3. **Pronoun match**: In the case of He_3^3 it is linked to $John_1^1$ cluster because of gender attribute. Gender attribute also helps to link her_{11}^{11} and $A\ girl_5^5$.

4. **Pronoun match**: It_7^7 is linked to $a\ new\ song_4^4$ based on the *animacy* attribute.

The multi-pass system approach also contemplates some post processing. For example, the singleton mention clusters are removed. They also removed, for the Ontonotes 4.0 corpus annotation the coreference links established in the *Precise Constructs* sieve shown in table 2.

5 Datasets and required NLP tools

As it has been seen so far, the way we have introduced the multi pass sieve coreference resolution system (Raghunathan et al., 2010; Lee et al., 2011) means that we have relatively few requirements for developing it for each language. There are two types of requirements: datasets and linguistic information needed to exploit the algorithms of each sieve pass and mention processing modules.

5.1 NLP tools

1. **Constituent Parser**: It is obvious that a *constituent parser* is required to apply the method presented here for each of the languages in OpeNER. Note that the parsing should provide the heads of at least the NPs.

2. **NER**: A named entity recognition system is also needed. This has already been developed within OpeNER for each language in WP3.

3. **POS**: A part of speech tagger is also required for each language. It has been agreed that this will be provided by WP6 for each OpeNER language.

4. **Static Dictionaries**: Mainly for pronoun resolution as referred in section 3.4. **Providing such resources is responsibility of each language-partner.**

5. **Hand-made heuristics or patterns** for mention detection as explained in section 2.1 **for each language**.

Therefore, as things stand now, we "only" need to have available a constituent parser for all six languages in OpeNER. It has been already listed in WP6 the various parsers available. However, it should also be noted that parsers can be trained for each language provided that a treebank is available[4]:

- Dutch: Alpino parser is trained on the Alpino Treebank.
- English: Parsers trained on Penn Treebank are easily accessible.

[4] `http://en.wikipedia.org/wiki/Treebank#List_of_treebanks_sorted_by_language`

- French: Vicomtech has already acquired a French Treebank and trained a parser.
- German: Several treebanks are listed although the *Negra* corpus has been used to train parsers (for example the Stanford Parser).
- Italian: There are several available, although the Turin University Treebank seems the most easily accessible[5]. Furthermore Synthema reported they can provide their own parser.
- Spanish: Ancora corpus[6]. Moreover, Freeling already provides parsing for Spanish.

5.2 Datasets

As explained in OpeNER deliverable D3.11, there are a number of evaluation datasets for coreference in the general domain:

1. **SemEval 2010**: provides datasets for every OpeNER language except French[7].
2. **CoNLL 2011**: provides dataset for English.

As the multi sieve coreference system is unsupervised we do not envisage the need of large annotated training data but rather we will require gold standards for evaluation in both the general domain and the tourist domain.

6 Roadmap

So, what now? WP3 needs to deliver by month 10 of the project, that means 30th of April, a coreference resolution system for all 6 languages in OpeNER. In principle, each partner is responsible for its language.

However, in order to make this easier, UPV/EHU will be providing the implementation functionalities required in the coreference module. Then each partner will have to adapt the module for their respective languages and will be responsible of its delivery in a github OpeNER repository. More specifically this means:

1. **Parsing**: Each partner needs to provide a parsing module that ideally provides outputs bracketed Penn Treebank format:

```
((S=H (NP (NP=H (NNP Pierre) (NNP=H Vinken))
(, ,) (ADJP (NP (CD 61) (NNS=H years)) (JJ=H old)) (, ,))
```

[5]http://www.di.unito.it/~tutreeb/index.html#treebank
[6]http://clic.ub.edu/corpus/ancora
[7]http://stel.ub.edu/semeval2010-coref/

2. **Static Dictionaries**: Each partner will need to provide the static dictionaries with the characteristics of the ones referred to in section 3.4.

3. **Hand-made patterns or heuristics for mention selection**: Each partner will need to rewrite such patterns for each language. Section 7 lists the patterns for pleonastic *it*, but these would need to be adapted for pleonastic pronouns in each language.

4. **Linguistic expertise** for mapping of syntactic and POS labels for mention processing.

5. **Linguistic expertise for designing sieves**. It should be noted that although we propose to implement all four sieve passes, the Exact String Match and the Pronoun Resolution sieve passes could be the priority to have a working system for the April deliverable.

Table 3 specifies the calendar towards a successful deliverable. The general idea is that, while UPV/EHU works on providing the software package to do the processing, we suggest that partners starting looking at the dictionaries and parsers needed for the coreference system to work in their languages, keeping in mind the dates as stated in table 3.

Date	Activity	Partners
30/03/2013	Parsers	All
15/03/2013	Basic Coreference module	UPV/EHU
30/04/2013	Patterns/Heuristics	All
30/04/2013	Static Dictionaries	All
30/04/2013	Language-dependent sieves	All
30/04/2013	Coreference Modules delivery	All

Table 3: Coreference Roadmap.

7 Annex A: Detecting Pleonastic 'It'

```
NP < (PRP=m1) $.. (VP < ((/^V.*/ < /^(?:is|was|become|became)/) $.. (VP < (VBN $.. /S|SBAR/))))
NP < (PRP=m1) $.. (VP < ((/^V.*/ < /^(?:is|was|become|became)/) $.. (ADJP $.. (/S|SBAR/))))
NP < (PRP=m1) $.. (VP < ((/^V.*/ < /^(?:is|was|become|became)/) $.. (ADJP < (/S|SBAR/))))
NP < (PRP=m1) $.. (VP < ((/^V.*/ < /^(?:is|was|become|became)/) $.. (NP < /S|SBAR/)))
NP < (PRP=m1) $.. (VP < ((/^V.*/ < /^(?:is|was|become|became)/) $.. (NP $.. ADVP $.. /S|SBAR/)))
NP < (PRP=m1) $.. (VP < (MD $ .. (VP < ((/^V.*/ < /^(?:be|become)/) $.. (VP < (VBN $.. /S|SBAR/))))))
NP < (PRP=m1) $.. (VP < (MD $ .. (VP < ((/^V.*/ < /^(?:be|become)/) $.. (ADJP $.. (/S|SBAR/))))))
NP < (PRP=m1) $.. (VP < (MD $ .. (VP < ((/^V.*/ < /^(?:be|become)/) $.. (ADJP < (/S|SBAR/))))))
NP < (PRP=m1) $.. (VP < (MD $ .. (VP < ((/^V.*/ < /^(?:be|become)/) $.. (NP < /S|SBAR/)))))
NP < (PRP=m1) $.. (VP < (MD $ .. (VP < ((/^V.*/ < /^(?:be|become)/) $.. (NP $.. ADVP $.. /S|SBAR/)))))
NP < (PRP=m1) $.. (VP < ((/^V.*/ < /^(?:seems|appears|means|follows)/) $.. /S|SBAR/))
NP < (PRP=m1) $.. (VP < ((/^V.*/ < /^(?:turns|turned)/) $.. PRT $.. /S|SBAR/)
```

References

Bergsma, S. and Lin, D. (2006). Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, page 3340.

Haghighi, A. and Klein, D. (2009). Simple coreference resolution with rich syntactic and semantic features. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing Volume 3 - EMNLP '09*, (August):1152.

Hobbs, J. R. (1978). Resolving pronoun references. *Lingua*, 44(4):311338.

Ji, H. and Lin, D. (2009). Gender and animacy knowledge discovery from web-scale n-grams for unsupervised person mention detection. *Proc. PACLIC2009*.

LDC, L. D. et al. (2005). *ACE (Automatic Content Extraction) English annotation guidelines for entities*. Version.

Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., and Jurafsky, D. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, pages 1–54.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D. (2011). Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, page 2834.

Poon, H. and Domingos, P. (2008). Joint unsupervised coreference resolution with markov logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, page 650659.

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., and Manning, C. (2010). A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, page 492501.

Weischedel, R., Pradhan, S., Ramshaw, L., Kaufman, J., Franchini, M., El-Bachouti, M., Xue, N., Palmer, M., Marcus, M., and Taylor, A. (2010). OntoNotes release 4.0. Technical report, Tech. rept. BBN Technologies.