Characterizing Mention Mismatching Problems for Improving Recognition Results

Jean Carlos Oliveira de Abreu
PPGCC/INE/CTC/UFSC
P.O. Box 476, Florianópolis-SC
Brazil
jean.abreu@posgrad.ufsc.br

Renato Fileto
PPGCC/INE/CTC/UFSC
P.O. Box 476, Florianópolis-SC
Brazil
r.fileto@ufsc.br

Axel-Cyrille Ngonga Ngomo
Institute for Applied Computer Science
Leipzig University
Germany
ngonga@informatik.uni-leipzig.de

Michael Röder
Institute for Applied Computer Science
Leipzig University
Germany
roeder@informatik.uni-leipzig.de

Matthias Wittwer
Information Systems Institute
Leipzig University
Germany
wittwer@wifa.uni-leipzig.de

Horacio Saggion
Department of Information and Communication Technologies
Pompeu Fabra University
Barcelona, Spain
horacio.saggion@upf.edu

ABSTRACT

Mentions to real world things which are recognized by software tools in text often mismatch the ground truth. This paper proposes a formal classification of mention mismatching problems, including partial matching. Then, it depicts evidence that some longer mentions are associated with higher precision and more specific things than shorter mentions that overlap them. Based on this, some algorithms are proposed to automatically improve mentions by increasing their sizes whenever and as much as possible. Experimental results applying a variety of state-of-the-art annotation tools against several datasets made from real world texts show that over-segmentation (returned mention contained in the corresponding one of the ground truth) is the most prevalent partial matching problem among those of the proposed classification. In addition, some of the proposed algorithms for mention enhancing were able to correct most over-segmented mentions returned by tools used in the experiments with prominent benchmarks, leading to gains in precision and recall.

CCS CONCEPTS

• **Applied computing - Annotation** • **Computing methodologies - Information extraction** • *Applied computing - Document analysis*.

iiWAS '17, December 4–6, 2017, Salzburg, Austria

ACM ISBN 978-1-4503-5299-4/17/12…$15.00
https://doi.org/10.1145/3151759.3151794

KEYWORDS

Text annotation, semantic annotation, NER, mention mismatch, segmented mentions.

1 INTRODUCTION

The vast amount of text currently available in digital media, including digital libraries, social media, and the Web in general, has a huge application potential. However, text data are considered unstructured for computational processing purposes, and their semantics can be vague and hard to grasp computationally. Thus, to completely realize the potential of textual data it is necessary to semantically enrich this data, i.e., to identify relevant portions of raw text, and to link them to some structured or semi-structured data which carry well-defined semantics (e.g. DBPedia[1] or Babelnet[2] resources). The resulting semantic annotations link relevant text portions (e.g., mentions to *D. Trump*, *The Trump Organization*, and *Trump Tower*) to resources describing them (e.g., a specific person, a company or a building, respectively). Such annotations help to elucidate meanings, allow semantic expansion, and enable automatic information processing and reasoning.

Named Entity Recognition (NER) [1] and Entity Linking (EL) [2] are popular tasks for semantically enriching text data by using techniques originated in areas such as Natural Language Processing (NLP), information extraction, and text mining. NER aims to identify each contiguous text portion (mention) that refers to some named entity, i.e., some concept or instance of concept such as person, organization, product, location, time, or currency

[1] http://wiki.dbpedia.org
[2] http://babelnet.org

amount. EL, by its turn, tries to link the mention to the exact named entity that it refers to, in a database or knowledge base.

Unfortunately, state-of-the-art techniques and tools for NER and analogous tasks sometimes fail to correctly identify mentions in text, what compromises subsequent tasks such as EL [3]. Table 1 gives some examples to illustrate the mention partial mismatching problem that we call over-segmentation. The supposedly correct mentions on the top of Table 1 (second line) are segmented into smaller sequences of terms (being each term a maximal sequence of alphanumeric characters), without loss (third line) or with loss (forth line).

Table 1: Examples of mention over-segmentation

Example 1	Example 2
[George H. W. Bush]	**[The Trump Organization]**
[George] [H.] [W.] [Bush]	**[The Trump] [Organization]**
[George] H. W. **[Bush]**	The **[Trump] [Organization]**

Over-segmentation can be harmful in certain cases, because it can hinder the correct linking of each mention to the right and specific thing that it refers to. For example, over-segmenting the name of a place that contains the name of a person (e.g. considering just the mention *Tom Jobim* instead of the whole mention *Tom Jobim Airport*) can lead to a quite different thing (*Tom Jobim*, the Brazilian musician who created the song *Girl from Ipanema*, instead of *Rio International Airport*). In the examples presented in Table 1, some smaller mentions ([*George*], [*Bush*], [*The Trump*], [*Trump*], and [*Organization*]) can cause disambiguation problems or misunderstandings. Notice that [*Organization*], for instance, is more general than [*The Trump Organization*] and just the mention *Trump* may refer to other things (e.g., card games, fictional characters, a magazine, some Islands in Antarctica).

Nevertheless, to the best of our knowledge, this and other segmentation mismatch problems have not been investigated in sufficient depth yet. Some works just sketch classifications for mention mismatching problems, including partial matches [3,15,16], but they miss many details or do not focus on improving recognized mentions. Sil & Yates [3] propose a method that combines NER with EL to generate more correct mentions. Some other proposals [4,5] intertwine NER with EL, aiming to improve mention recognition performance, and EL results as a consequence. Other proposals employ dictionaries [6–11] or combine results of distinct tools [8,9,12,13] to improve NER and EL results. Most of the approaches discussed above frequently end up by increasing mention size, maybe not intentionally and as a side effect, when trying to improve results. However, none of these works provides formal definitions for mention mismatching classes, particularly for partial matches, and they do not explicitly exploit such classes, their incidences or mention size maximization to improve mention recognition.

This work investigates mention mismatching problems i.e., differences between mentions in some ground truth and the ones automatically found in text by software tools, propose a solution for some of the classes of these problems that we call over-segmentation, and assess how our solution contributes for improving mention recognition in experiments. Thus, the major contributions of this paper are: (i) a formal classification of segmentation mismatching problems, including partial matches (when a mention of the ground truth partially overlaps mention(s) returned by a tool); (ii) a family of algorithms called MInT (*Mention Increasing in Text*) that expand some mentions to correct over-segmentation; and (iii) extensive experimental results that validate our approach with a variety of state-of-the-art tools and several prominent datasets built on real world text, which have frequently been used in the literature to compare the performance of NER and EL tools.

Our classification of mention partial matches enables analysis of the incidence of variations of these problems in different case studies and datasets. It leads to more insights about these problems and their possible solutions, to better improve mention recognition results than just considering general performance measures, such as precision and recall. Our MInT algorithms can be used to correct over-segmented mentions found by a variety of tools, including NER tools and tools that recognize in text mentions to names of things present in a database or knowledge base. In the MInT prototype, these algorithms run as alternative post-processing steps that improve recognized mentions, to facilitate the assessment of their performance with different tools and datasets. However, MInT strategies for improving mentions can be incorporated in the tools themselves for more efficient processing.

The experimental results with a corpus in Portuguese and a variety of corpuses available for experiments using the Gerbil framework [16] show that over-segmentation is more prevalent than other mention mismatching problems in the results of several annotator against several datasets, and our method was able to correct over-segmentation in most cases. The mention recognition results corrected with our method have presented higher F-measure than the bare results of annotators in many cases.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 presents our formal classification of mention mismatching problems. Section 4 describes our solution for some mention mismatching problems. Finally, Section 5 reports and discusses experimental results, and Section 6 concludes the paper, with final remarks and an enumeration of issues for future work.

2 RELATED WORK

Cornoli *et al.* [15] and later Röder *et al.* [16] propose to take into account what they call *weak annotation matching* to assess and compare the performance of semantic annotation tools. However, their conditions for *weak annotation matching* just relax the matching criteria, by allowing to count as a match any generated mention that overlaps a mention of the ground truth. Sil & Yates [3] informal classification of mention mismatching problems is based just on examples, and does not cover all the topological relation possibilities between mentions. Its incompleteness and lack of formal definitions makes this classification difficult to grasp and use. They also propose a method that employs a set of candidate mentions identified by an NER tool and a set of selected

surface names selected via EL to perform a ranking of (mention, surface name) pairs, and produce a prediction model. Luo et al. [4] considers the mutual dependency between NER and EL, in which the decision of entity linkage in EL influences the identification of mentions in NER. Nguyen et al. [5] also combines NER and EL, by using a probabilistic graphical model to capture mention spans, mention types, and the mapping of mentions to entities in a knowledge base.

Dictionary-based methods are widely used for comparing text substrings (candidate mentions) with dictionary strings. Tools for recognizing mentions and linking them to DBPedia resources, such as Wikify [6] and DBPedia Spotlight [7], can be considered dictionary-based. Although these tools use a longest size strategy for comparing text substrings (candidate mentions) with dictionary strings, they can generate a number of over-segmented mentions, as shown in Section 5.

There are also methods that combine NER with PoS-Tagging and dictionary tools for better recognition and classification of mentions in text. Chiu et al. [8] exploits the longest mention strategy, in a method combining a dictionary with DBPedia-Spotlight and TAGME. However, their computational cost can be high, because the text is entirely traversed by many tools. Gamallo & Garcia [9] proposes a method that receives a list of candidate mentions identified with the help of a PoS-Tagging tool, and uses a dictionary, derived from Wikipedia, to carry out the classification of these mentions into categories, such as person and organization. However, they do not consider mention mismatching problems.

Text similarity measures have been used to partially match text portions that can be mentions with surface names of a dictionary. The method proposed by Li et al. [10] extracts n-grams from the text, and compare them with dictionary strings using the edit distance similarity. Then, they select the similar strings with maximum size. Deng et al. [11] extend [10] to build a unified framework with support for several similarity measures (edit distance, token distance, etc.) for comparing n-grams from the text with surface names in a dictionary.

Plu et al. [12] proposes a method with three pipelines executed over the text. Pipeline 1 selects proper names, pipeline 2 uses Stanford NER to recognize named entities, and pipeline 3 combines the results of the previous pipelines. Their method weaves partially overlapping mentions identified by their respective tools. Despite treating overlapping, terms not contained in the mentions recognized by previous pipelines are discarded, what inhibits mention maximization. FOX [13] apply ensemble learning to combine the results of some of the most prominent NER and EL tools, and uses AGDISTIS [14] to disambiguate entities on linked databases and achieve higher precision and recall. Nevertheless, its final results can include a number of over-segmented mentions yet.

To the best our knowledge, this work is the first to provide formal definitions of mention mismatching classes which help understand and correct mention mismatching problems. In addition, our method for correcting mentions only performs comparisons between mentions identified in the text by one tool and strings present in the dictionary. It traverses only a relatively small portion the text around the mentions identified by NER tools. The sliding window size is calculated according to the surface names found in the dictionary to contain the mention to be corrected. Thus, it is more computational efficient than many proposals in the literature. Finally, our experimental results prove the effectiveness and the computational efficiency of our method, besides showing the higher prevalence of over-segmentation in comparison with other classes of mention mismatching problems in a real world case study.

3 CLASSES OF MENTION MISMATCHES

This section provides formal definitions for classes of mismatches between mentions of a ground truth and mentions automatically found by a mention recognition tool. The mention mismatching classes that we propose are based on those informally described (by using just examples) in the work of Sil & Yates [3]. However, we have no commitment with perfect compatibility with the classes described in that paper.

Let *GT* (Ground Truth) be a set of mentions correctly recognized in a natural language text *T*, and *I* a set of mentions found by some method or tool in the same text *T*. Given two mentions $GTM_i \in GT$ and $IM_j \in I$, we say that GTM_i *perfectly matches* IM_j if they refer to exactly the same portion of the text *T* (with respect to their limits in *T*, not just their textual contents). We say that GTM_i *does not overlap* IM_j if the portions of *T* that they refer to do not have any sub-portion in common. Any $IM_j \in I$ which does not overlap any $GTM_i \in GT$ (false positive) is usually called a *spurious mention* in the literature. Conversely, any $GTM_i \in GT$ which does not overlap any $IM_j \in I$ (false negative) is called an *unmatched mention*. Of course, spurious mentions compromise precision, and unmatched mentions compromise recall.

In this paper, besides perfect matching, spurious mentions, and unmatched mentions, we are interested in partial matching. We say that $GTM_i \in GT$ partially matches (partially overlaps) an $IM_j \in I$ if GTM_i, and IM_j are different but share some portion of the text *T*, i.e., $GTM_i \neq IM_j$ AND ($GTM_i \sqsupset IM_j$ OR $GTM_i \sqsubset IM_j$)[3]. The following sub-sections formally describe our classes of mention mismatching problems bases on partial matching, and provide examples of the respective classes.

3.1 Over-segmentation

Over-segmentation occurs when a ground truth mention $GTM_i \in GT$ has no perfect matching with any mention in the set *I* returned by a tool, but partially matches a number of mentions $I' \subseteq I$. In other words, each mention $IM_j \in I'$ refer to a portion of the text *T* contained in GTM_i. Definition 3.1 formally states this phenomenon.

[3] In this paper, the squared operators for the containment predicates ($\sqsubset, \sqsupset, \sqsubseteq, \sqsupseteq, \not\sqsubseteq$), the intersection predicate ($\sqcap$), and the composition of mentions ($\sqcup$) denote the respective operations on portions of a text *T* that constitute the mentions used as arguments of these operators. These mentions are determined with respect to their limits in *T*, and not just their text contents, because the same textual content (string) can appear repeatedly in the same text.

Definition 3.1 **(Over-Segmented Mention):** Given two sets of mentions *GT* (Groud True) and *I* (Identified by some method) in the same text *T*, one mention $GTM_i \in GT$ is *over segmented* if *I* does not have any mention that perfectly matches GTM_i, but has a subset of mentions $I' \subseteq I$ ($|I'| \geq 1$) such that $IM_j \sqsubset GTM_i$ for all $IM_j \in I'$.

Figure 1 illustrates cases of over-segmentation and its subclasses, namely over-segmentation without loss and over-segmentation with loss. Over-segmentation without loss (Figure 1 (a)) refers to any case of over-segmentation in which the concatenation of the mentions in I´ (the ones of *I* that partially overlap GTM_i) is equal to GTM_i. In over-segmentation with loss (Figure 1 (b)), on the other hand, the composition of the mentions in *I´* leave one or more gaps, i.e., portions of GTM_i that are not contained in any mention of *I'*.

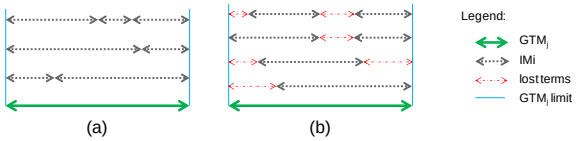


Figure 1: Over-segmentation without loss (a), and with loss (b)

Example **3.1**: The segmentation of the mention *[George H. W. Bush]* in the smaller mentions *[George]*, *[H.]*, *[W.]*, and *[Bush]* constitutes over-segmentation without loss, while its segmentation contemplating only the mentions *[George]* and *[Bush]* constitutes over-segmentation with loss, because it leaves a gap with the terms *H.* and *W.* Analogously, considering the complete mention *[The Trump Organization]*, the smaller mentions *[Trump]* and *[Organization]* constitute an over-segmentation without loss, while just the mention with one term *[Organization]* constitutes an over-segmentation with loss.

3.2 Under-segmentation

Under-segmentation occurs when a ground truth mention $GTM_i \in GT$ just partially matches one or more mentions *I* returned by a tool. In other words, *I* do not have any mention that perfectly matches GTM_i but has a subset $I' \subseteq I$ of mentions partially overlap GTM_i and that also include portions of the text *T* not included in GTM_i. Definition 3.2 formalizes this phenomenon.

Definition 3.2 **(Under-segmented Mention):** Given two sets of mentions *GT* (Groud True) and *I* (Identified by some method), in the same text *T*, one mention $GTM_i \in GT$ is *under segmented* if *I* does not have any mention that perfectly matches GTM_i, but has a subset of mentions $I' \subseteq I$ ($|I'| \geq 1$) such that $IM_j \not\sqsubseteq GTM_i$ and $IM_j \sqcap GTM_i \neq \varepsilon$ for all $IM_j \in I'$.

Example **3.2**: One tool recognized the mention of an address in Portuguese, *[Av. Auro Soares de Moura Andrade, 664]*, while the ground truth was just the mention referring to the avenue name *[Av. Auro Soares de Moura Andrade]*. This characterizes under-segmentation without loss. Notice that the under-segmented mention refers to a more specific thing than the ground truth mention in this case. On the other hand, if the returned mention had been *[Auro Soares de Moura Andrade, 664]*, it would characterize an under segmentation with loss, because despite of the presence of the number, the term *Av.* (avenue), present in the ground truth mention, would be missing.

Figure 2 illustrates cases of over-segmentation and its subclasses. Under-segmentation without loss (Figure 1 (a)) does not leave gaps, while under-segmentation with loss (Figure 1 (b)) does. In other words, in under-segmentation without loss the composition of the segments in *I´* completely covers GTM_i while in under segmentation with loss some terms of GTM_i do not appear in any mention of *I'*.

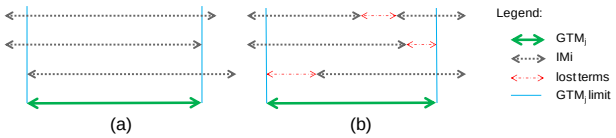


Figure 2: Under-segmentation without loss(a), with loss(b)

3.3 Mixed-Segmentation

Mixed-segmentation is a mixture of at least one case of over-segmentation and at least one case of under-segmentation with respect to the same ground truth mention $GTM_i \in GT$. Definition 3.3 formalizes this phenomenon illustrated in Figure 3.

Definition 3.3 **(Mixed-Segmented Mention):** Given two sets of mentions *GT* (Groud True) and *I* (Identified by some method) in the same text *T*, one mention $GTM_i \in GT$ is *mixed segmented* if *I* does not have any mention that perfectly matches GTM_i, but there is a subset of mentions $I' \subseteq I$ ($|I'| \geq 1$) such that $IM_j \sqsubset GTM_i$ for all $IM_j \in I'$, and another subset of mentions $I'' \subseteq I$ ($|I''| \geq 1$) such that $IM_j \not\sqsubseteq GTM_i$ and $IM_j \sqcap GTM_i \neq \varepsilon$ for all $IM_j \in I''$.

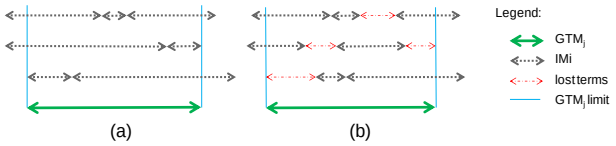


Figure 3: Mixed-segmentation without loss (a), and with loss (b)

Example 3.3: Though the ground truth for the text passage in Portuguese *Nobel de Literatura Octavio Paz* considered just the mention *[Nobel de Literatura]*, a tool returned the mentions *[Nobel]* and *[Literatura Octavio Paz]*. It characterizes mixed segmentation with loss, because it involves an over-segmentation of the ground truth (*[Nobel]*), an under-segmentation (*[Literatura Octavio Paz]*), and a gap (the absence of the preposition de in the results returned by the tool). Notice that, once again in this case, the longest mention *[Nobel de Literatura Octavio Paz]* would refer to the most exact and specific thing, despite the shorter ground truth and the shorter mentions returned by the tool.

3.4 Classification Summary

Table 2 summarizes the classes of mention mismatching problems involving partial matching between a ground truth mention GTM_i and a number of the mentions in the same text *T* returned by some

tool, each one denoted by $IM_j \in I$. The right column of Table 2 presents the rule that defines the respective mention partial matching class in the left column. Remember that *I'* denotes a set of mentions of *I* that partially match GTM_i. The incidence of these classes of partial mismatches in experiments using state-of-the-art tools and a variety of datasets is presented in Section 5.1.

Table 2: Mention mismatch classification summary

Class	Rule
Over-segmentation	$IM_i \sqsubset GTM_i$
With loss	$\left(\coprod_{IMj \in I'} IMj\right) \sqsubset GTM_i$
Without loss	$\left(\coprod_{IMj \in I'} IMj\right) = GTM_i$
Under-segmentation	$IM_i \not\subseteq GTM_i$, $IM_i \sqcap GTM_i \neq \varepsilon$
With loss	$\left(\coprod_{IMj \in I'} IMj\right) \not\supseteq GTM_i$
Without loss	$\left(\coprod_{IMj \in I'} IMj\right) \sqsupset GTM_i$

4 THE MInT METHOD

The MInT (Mention Increasing in Text) method can be seen as a post-processing step (or, to avoid overhead, a strategy implemented in some existing mention recognition tool) for solving over-segmentation problems through mention expansion to the surrounding text. It can be done by a variety of alternative algorithms, such as the ones that we have developed based on dictionaries, and describe in the following.

Our algorithms MInT Naïve, MInT NoIn and MInT NoOver rely on any dictionary of surface names for driving the mention expansions. Our dictionary-based approach for mention expansion allows special characters and punctuation signs in surface names, what usually has a negative impact on the performance of mention recognition tools [1]. MInT Naïve just exchanges each mention returned by a tool with the longest surface name that matches that mention and its surrounding text, if there is such a surface name in the dictionary. MInT NoIn and MInT NoOver also try to maximize mention length according with a dictionary of surface names and the surrounding text, but avoid returning expanded mentions that are contained in other mentions or overlap other mentions, respectively. By doing this, these two algorithms are able to correct over-segmented mentions without causing many cases of under-segmentation as a side effect. MInT NoIn and MInT NoOver, our algorithms that presented the best results in experiments, are presented in the following subsections. Their performance evaluation with a variety of corpuses and tools is presented in Section 5.2.

4.1 MInT NoIN

MInT NoIn (Algorithm 1) takes as inputs the same arguments as MInT Naïve, namely a text document *D*, a list *IM* of mentions previously recognized in *D*, and a list of surface names in decreasing order of their size. It returns as outputs a list *MM* of maximized mentions (i.e. the longest surface names matching some mentions in *IM* and their surrounding text), a list of mentions $IMsNotFoundInSN \subseteq MM$ that do not match any surface name in *SN*. And in addition to the output parameter of MInT Naïve, a list of mentions *RM* removed from *IM* to eliminate maximized mentions contained in or duplicating other mentions. However, MInT NoIn allows partially overlapping mentions in its results. The combination of *MM*, *IMsNotFoundInSN* and *RM* is of the size of the mentions list *IM* provided as input (i.e., $|MM \cup IMsNotFoundInSN \cup RM| = |IM|$).

Algorithm 1 MInT NoIn – Maximize mentions removing duplicates

```
Input: D, // Text document
       IM, // List of mentions found in D by some tool ordered by offset
       SN; // List of surface names in decreasing size order
Output: MM, // List of maximized mentions
        IMsNotFoundInSN; // List of mentions not found in SN
        RM; // List of mentions removed from IM due to duplicates caused by maximization of mentions
1.  lengthD = getLength(D);
2.  MM = [];
3.  RM = [];
4.  IMsNotFoundInSN = [];
5.  offsetLastIM = -1;
6.  i = 0;
7.  FOREACH IMj IN IM DO
8.    IMExistInSN = false;
9.    FOREACH sn IN SN DO
10.     IF sn.length > IMj.length THEN
11.       offsetIMinSN = getOffset(IMj.label, sn.label)
12.       IF offsetIMinSN >= 0 THEN
13.         ts = GetTS(IMj.label, sn.label, D)
14.         offsetSNinTS = getOffset(sn.label, ts.label)
15.         IF offsetSNinTS >= 0 THEN
16.           IMj.offset = ts.offset + offsetOfSNinTS;
17.           IMj.label = sn.label;
18.           IMj.length = sn.length;
19.           MM.push(IMj);
20.           IMExistInSN = true;
21.     ELSEIF sn.length == IMj.length THEN
22.       IF sn.label == IMj.label THEN
23.         IMExistInSN = true;
24.         break;
25.     ELSE
26.       break;
27.   DONE
28.   IF (IMj.offset = offsetLastIM) THEN // if duplicate
29.     RM.push(IMj);
30.     IMj.remove(i); i
31.   IF IMExistInSN == false THEN
32.     IMsNotFoundInSN.push(IMj);
33.   offsetLastIM = IMj.offset;
34.   i = i + 1;
35. DONE
36. RETURN(<IM, MM, IMsNotFoundInSN>)
```

4.2 MInT NoOver

MInT NoOver (Algorithm 3) takes the same inputs and returns as outputs the same set of output parameters as MInT NoIn. The difference between these algorithms is that MInT NoOver removes any maximized mention that overlaps (not just the ones

contained in) other mentions, and returns all the removed mentions in the output parameter *RM*.

Algorithm 2: *MInT NoOver* – Maximizes mentions removing overlaps

```
Input: D, // Test document
       IM, // List of mentions found in D by some tool ordered by offset
       SN; // List of surface names in decreasing size order
Output: MM, // List of maximized mentions
        IMsNotFoundInSN; // List of mentions not found in SN
        RM; // List of mentions removed from IM due to duplicates caused by maximization of mentions
1.  lengthD = getLength(D);
2.  MM = [];
3.  RM = [];
4.  IMsNotFoundInSN = [];
5.  lastIM = NULL;          // last processed IM
6.  i = 0;
7.  FOREACH IMj IN IM DO
8.     IMExistInSN = false;
9.     FOREACH sn IN SN DO
10.       IF sn.length > IMj.length THEN
11.          offsetIMinSN = getOffset(IMj.label, sn.label)
12.          IF offsetIMinSN >= 0 THEN
13.             ts = GetTS(im.label, sn.label, D)
14.             offsetSNinTS = getOffset(sn.label, ts.label)
15.             IF offsetSNinTS >= 0 THEN
16.                IMj.offset = ts.offset + offsetOfSNinTS;
17.                IMj.label = sn.label;
18.                IMj.length = sn.length;
19.                MM.push(IMj);
20.                IMExistInSN = true;
21.       ELSEIF sn.length == im.length THEN
22.          IF IMj.label == IMj.label THEN
23.             IMExistInSN = true;
24.             break;
25.       ELSE
26.          break;
27.    DONE
28.    IF (IMj.offset <= lastIM.end) THEN // current IM overlaps last processed IM
29.       treatOverlap(i, IMj, lastIM, RM, IM);
30.    IF IMExistInSN == false THEN
31.       IMsNotFoundInSN.push(IMj);
32.    IMj.end = IMj.offset + im.length;
33.    lastIM = IMj;
34.    i = i + 1;
35. DONE
36. RETURN(<IM, MM, IMsNotFoundInSN>)
```

5 EXPERIMENTAL EVALUATION

The experiments to validate our proposal aim to analyze the incidences of the classes of mention mismatching problems defined in Section 3, and the benefits of the algorithms proposed in Section 4 to solve over-segmentation. The first experiments, reported in Section 5.1, applied Priberam REM [10,4] followed by MInT to the dataset Golden Collection (CD-2) of the second HAREM (event occurred in 2008). The other experiments, reported in Section 5.2, applied a variety of state-of-the-art annotators to prominent datasets, all of them available and integrated in the Gerbil[4] framework [16].

HAREM CD-2 is one of the biggest and most used benchmarks for NER in Portuguese. Its corpus has 466,355 words, and was created from journalistic, literary, and political texts, found on the Web, being some of them transcribed from interviews. All the texts present on CD-2 were annotated and checked by humans, to produce a ground truth. The Priberam REM entity recognition tool [17] was used to recognize mentions in the HAREM CD-2 corpus because it obtained the best levels of precision and recall for mention recognition in Portuguese texts, and identified the greatest number of mentions in the HAREM CD-2 corpus. Gerbil has been chosen for further experiments because it encompasses the largest and most current collection of datasets and annotators. In addition, it relies on current standards to facilitate connection of datasets and annotators, and help manage and analyze experimental results.

5.1 Priberam REM on HAREM CD-2

These experiments enhanced the mentions recognized by Priberam REM on HAREM CD-2 by using MInT with two distinct dictionaries of surface names: GT with all the 3,482 ground truth mentions of HAREM CD-2, and DP with 1,667,261 surface names (values of the property label) taken from DBPedia in Portuguese. GT was used to verify if MInT is able to correct over-segmented mentions at least with an ideal dictionary, as well as to compare the performance of different MInT algorithms to handle overlaps caused by mention enhancement.

Priberam REM recognized 6,434 mentions in the HAREM CD-2 corpus. These mentions were aligned with the 5,641 ground truth mentions by using the "Aligner" program provided with the dataset. Table 3 summarizes the observed incidences of mention mismatches in absolute numbers and as a percentage of the number of ground truth mentions (between parenthesis). Notice the high proportion of partial matches in comparison with other mention mismatch problems (spurious and unidentified mentions). Both MInT NoIn and MInT NoOver solved most of these problems by using GT, and a much smaller proportion of them by using DP, as expected. Notice the post-processing using MInT NoOver resulted in a slightly smaller number of partial matches (121 for GT, and 626 for DP) than MInT NoIn GT (123 for GT, and 635 for DP), due to the elimination of mentions contained in maximized mentions.

Table 3: Results summary for Priberam REM on HAREM CD-2

	Priberam REM	MInT NoIn GT	MInT NoIn DP	MInT NoOver GT	MInT NoOver DP
Recognized	6434	5799	6228	6200	6200
Spurious	288 (4.5%)	282 (4.9%)	287 (4.6%)	282 (4.9%)	287 (4.6%)
Non-recognized	100 (1.7%)	100 (1.81%)	100 (1.7%)	109 (1.9%)	110 (2%)
Correctly identified	4709 (83.5%)	5416 (96%)	4906 (86.9%)	5411 (95.9%)	4905 (87%)
Partial matches	**832 (14.7%)**	**123 (2.2%)**	**635 (11.3%)**	**121 (2.1%)**	**626 (11.1%)**

Table 4 details the incidences of partial matches in mentions recognized by Priberam REM alone, and in the same mentions

[4] http://aksw.org/Projects/GERBIL.html

enhanced by MInT NoOver DP with the two dictionaries (GT and DP). For each one, Table 4 presents the absolute number of each kind of partial match (#), and the percentages that these numbers represent of the total namber of ground truth mentions (%GTM), and the total number of partial matches (%PM). Notice that the total number of over-segmentations (with loss + without loss) in the bare Priberam REM results was more than 6 times greater than the total number of the other classes of partial matches. Notice that MInT NoOver GT corrected 699 cases of over-segmentation, leaving only 3 mentions over-segmented. These remaining over-segmentation cases are due to problems in the texts, such as extra spaces and the use of lower case letters when they should be upper case, as we observed in experiments. In cases of under-segmentation there was no correction, as expected, but just a slight negative impact. In cases of mixed segmentation (15 cases), only over-segmented mentions were maximized. It contributed to slightly increase the number of under-segmentations. Only 2 mentions went from correct to incorrect, increasing the total number of under-segmentation cases from 115 to 118. MInT NoOver using DP, on the other hand, solved just 238 over-segmentation cases and 5 cases of mixed segmentation, increasing the number of under-segmentation by 37 cases.

Table 4: Partial matches for Priberam REM on HAREM CD-2

Partial Match	Priberam REM			MInT NoOver GT			MInT NoOver DP		
	#	%GTM	%PM	#	%GTM	%PM	#	%GTM	%PM
Over-segm.	***702***	***12.4***	***84.4***	***3***	***0.1***	***2.5***	***464***	***8.2***	***74.1***
with loss	*561*	*9.9*	*67.4*	*2*	*0*	*0.8*	*389*	*6.9*	*62.1*
without loss	*141*	*2.5*	*16.9*	*1*	*0*	*1.7*	*75*	*1.3*	*12*
Under-segm.	***115***	***2***	***13.8***	***118***	***2.1***	***97.5***	***152***	***2.7***	***24.3***
with loss	*8*	*0.1*	*1*	*7*	*0.1*	*5.8*	*17*	*0.3*	*2.7*
without loss	*107*	*1.9*	*12.9*	*111*	*2*	*91.7*	*135*	*2.4*	*21.6*
Mixed-segm.	***15***	***0.3***	***1.8***	***0***	***0***	***0***	***10***	***0.2***	***1.6***
with loss	*12*	*0.2*	*1.4*	*0*	*0*	*0*	*10*	*0.2*	*1.6*
without loss	*3*	*0.05*	*0.4*	*0*	*0*	*0*	*0*	*0*	*0*

Table 5 summarizes the gains obtained by the application of MInT algorithms to the mentions identified by Priberam. F-Measure increased by 16.69%, precision by 20.21% and recall by 12.54% com MInT NoIn GT; and 4.85%, 5.93% and 3.48% respectively with MInT NoOver DP. This shows that MInT algorithms can correct cases of over-segmentation and mixed-segmentation, to the point of improving the overall quality of results when used with a good dictionary of surface names.

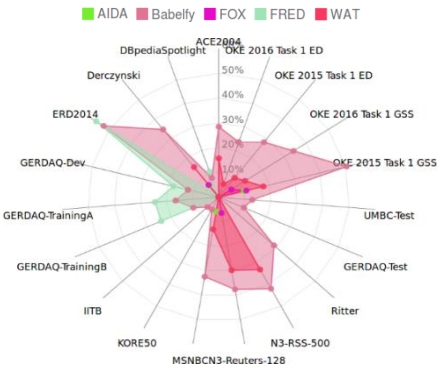
Table 5: MInT NoIn gains on Priberam REM for HAREM CD-2

	Priberam REM	MInT NoIn GT	MInT NoIn DP	MInT NoOver GT	MInT NoOver DP
Precision	*73.18 %*	*93.39 %*	*78.77 %*	***93.42 %***	*79.11 %*
Recall	*83.47 %*	***96.01 %***	*86.97 %*	*95.92 %*	*86.95 %*
F-Measure	*77.99 %*	***94.68 %***	*82.66 %*	*94.65 %*	*82.84 %*

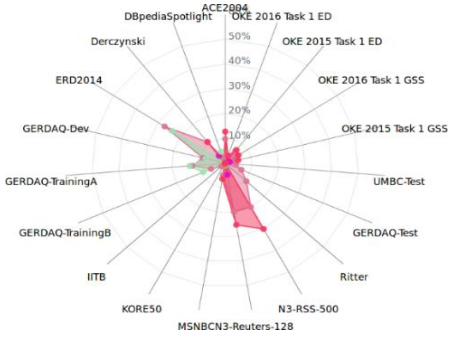
5.2 Gerbil Annotators and Datasets

We have extended Gerbil to calculate the incidences of the mention mismatch problems described in Section 3, and to allow the use of MInT as a post-processing step to expand mentions. The experiments realized in this extended version of Gerbil used a dictionary containing all the surface names taken from the label[5] properties of resources in the DBPedia [6] dataset, making 12,845,172 surface names, originaly from titles of Wikipedia articles and anchors of article links in Wikipedia disambiguation pages. More than 87 experiments have been realized with this dictionary on Gerbil so far, each one with a distinct annotator and dataset. However, due to space limitation, only the results of 40 pair annotator-dataset with the highest number of over-segmentation are presented in this paper.

Figure 4 shows the decrease in the number of cases of over-segmentation propitiated by using MInT NoOver with the annotators and datasets that generated the highest number of over-segmentations. The reduction is particularly high (from 57% to 13%) for the annotator Babelfy [18] in dataset OKE 2015 Task GSS, but not with Fred [19], which also has a high incidence of over-segmentation for some datasets. Figure 5 shows the increase in the number of cases of under-segmentation caused by MInT NoOver as a side effect. Nevertheles, it is partially due to the resolution of over-segmentation in cases of mixed-segmentation.



(a) Over-Segmentation without MInT



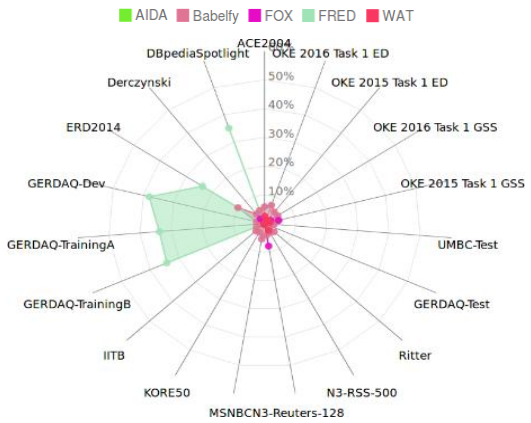
(b) Over-segmentation after MInT NoOver

Figure 4: Diminishing over-segmentation cases by using MInT

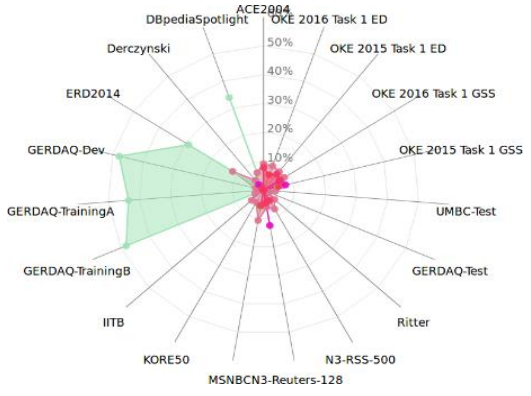
Table 6 presents the total number of partial matches (columns Over, Under, and Mixed, referring to over-segmentation, under-segmentation, and mixed-segmentation, respectively) found in the mentions recognized by annotators on

[5] http://wiki.dbpedia.org/services-resources/documentation/datasets#Labels
[6] http://wiki.dbpedia.org/downloads-2016-10

datasets (first column) that allowed the greatest number of resolutions of over-segmentation cases with Mint. Its remaining columns show the changes in the number of cases of over-segmentation (δO), under-segmentation (δU), and mixed-segmentation (δM) after applying MInT NoIn and MinT NoOver to enhance those mentions. Of course, the reductions tend to be bigger when the there are more cases of over-segmentation to correct. These corrections tend to be accompanied by proportional increases in the number of cases of under-segmentation. However, as can be observed in the last row of Table 6, the total reduction in over-segmentation cases is more than twice the increase in under-segmentation cases. In addition, occurences of mixed-segmentation cases are also reduced.



(a) Under-Segmentation without MInT



(b) Under-Segmentation after MInT NoOver

Figure 5: Increases in under-segmentation due to MInT

Table 7 presents the percentage of the changes in the number of over-segmentations (δ %O), under-segmentations (δ %U) and mixed segmentations (δ %M) obtained by applying MInT NoIn and MinT NoOver to enhance the mentions reconized by the same annotators and datasets listed in Table 6. MInT NoIn reduced over-segmentation cases by 8.7%, what is almost 4 times the 2.3% increase in cases of under-segmentation. MInT NoOver, by its turn, reduced over-segmentation cases by 11.7%, but causing a higher increase of 3.1%, which is also around a fourth of the gains in over-segmentation.

Table 6: Effect of MInT NoIn on partial match incidences

Annotator / Dataset	Without MInT			MInT NoIn			MInT NoOver		
	Over	Under	Mixed	δO	δU	δM	δO	δU	δM
Babelfy / IITB	1075	698	112	-666	277	-91	-728	265	-103
Babelfy / OKE 2016 Task 1 GSS	358	53	28	-295	43	-27	-303	31	-28
Babelfy / Ritter	429	63	4	-236	12	-3	-268	9	-4
Babelfy / N3-RSS-500	420	36	23	-213	41	-19	-219	39	-19
Babelfy / MSNBC	243	41	22	-202	43	-21	-207	39	-21
Babelfy / UMBC-Test	287	69	10	-182	20	-6	-198	18	-8
Babelfy / OKE 2015 Task 1 GSS	172	15	10	-155	14	-10	-159	7	-10
Babelfy / N3-Reuters-128	332	36	4	-154	15	4	-157	17	1
Babelfy / OKE 2015 Task 1 ED	187	34	15	-142	25	-14	-146	20	-15
Babelfy / OKE 2016 Task 1 ED	80	23	1	-69	8	-1	-70	7	-1
Babelfy / Derczynski	99	12	1	-61	0	0	-68	0	-1
FRED / GERDAQ-TrainingB	104	148	42	-14	16	24	-65	61	-24
WAT / OKE 2016 Task 1 GSS	126	19	0	-64	50	0	-64	48	0
Babelfy / ACE2004	87	18	7	-53	11	-4	-57	10	-5
WAT / MSNBC	98	3	0	-48	38	0	-50	38	0
FRED / GERDAQ-TrainingA	108	149	57	0	0	0	-49	43	-45
FOX / OKE 2016 Task 1 GSS	60	23	2	-46	42	-2	-46	41	-2
FRED / GERDAQ-Dev	75	160	62	0	0	0	-43	41	-42
WAT / OKE 2015 Task 1 GSS	60	6	0	-43	13	0	-43	11	0
WAT / N3-Reuters-128	263	0	0	-41	41	18	-41	40	18
AIDA / MSNBC	44	4	0	-35	39	0	-35	39	0
FOX / OKE 2015 Task 1 GSS	37	16	2	-31	10	-2	-31	9	-2
WAT / N3-RSS-500	332	27	5	-30	14	2	-30	14	2
AIDA / OKE 2015 Task 1 GSS	34	5	0	-29	11	0	-29	11	0
Babelfy / GERDAQ-TrainingB	46	12	1	0	0	0	-21	1	-1
WAT / OKE 2015 Task 1 ED	66	9	0	-21	37	0	-21	37	0
FRED / DBpediaSpotlight	36	116	11	-9	2	-1	-20	-4	-11
Babelfy / DBpediaSpotlight	27	16	0	-19	5	0	-20	5	0
FRED / ERD2014	33	14	5	-2	-4	6	-19	3	-4
Babelfy / GERDAQ-TrainingA	73	6	1	0	0	0	-18	0	0
Babelfy / GERDAQ-Test	43	8	0	0	0	0	-16	0	0
Babelfy / ERD2014	31	6	1	0	0	0	-15	1	-1
Babelfy / GERDAQ-Dev	51	11	1	0	0	0	-15	0	0
FOX / N3-Reuters-128	57	70	1	-15	41	0	-15	40	0
WAT / Derczynski	44	0	0	-13	2	0	-13	2	0
WAT / ACE2004	48	8	0	-9	16	1	-9	16	1
AIDA / Derczynski	17	0	0	-9	1	0	-9	1	0
WAT / OKE 2016 Task 1 ED	19	1	0	-8	18	0	-8	18	0
FOX / Derczynski	18	6	0	-7	1	0	-7	1	0
Babelfy / KORE50	8	5	0	-7	3	0	-7	2	0
Legend: GSS = Gold Standard Sample ED = Evaluation Dataset			Total:	-2928	905	-146	-3339	981	-325

Table 7: Percentages of reductions in partial matches with MInT

Annotator / Dataset	GTMs	Without MInT			MInT NoIn			MInT NoOver		
		%O	%U	%M	δ %O	δ %U	δ %M	δ %O	δ %U	δ %M
Babelfy / OKE 2015 Task 1 GSS	341	50,4	4,4	2,9	-45,5	4,1	-2,9	-46,6	2,1	-2,9
FRED / ERD2014	59	55,9	23,7	8,5	-3,4	-6,8	10,2	-32,2	5,1	-6,8
Babelfy / OKE 2016 Task 1 GSS	1049	34,1	5,1	2,7	-28,1	4,1	-2,6	-28,9	3,0	-2,7
Babelfy / MSNBC	747	32,5	5,5	2,9	-27,0	5,8	-2,8	-27,7	5,2	-2,8
Babelfy / ERD2014	59	52,5	10,2	1,7	0,0	0,0	0,0	-25,4	1,7	-1,7
Babelfy / Derczynski	286	34,6	4,2	0,3	-21,3	0,0	0,0	-23,8	0,0	-0,3
Babelfy / OKE 2015 Task 1 ED	664	28,2	5,1	2,3	-21,4	3,8	-2,1	-22,0	3,0	-2,3
Babelfy / N3-RSS-500	1000	42,0	3,6	2,3	-21,3	4,1	-1,9	-21,9	3,9	-1,9
Babelfy / OKE 2016 Task 1 ED	340	23,5	6,8	0,3	-20,3	2,4	-0,3	-20,6	2,1	-0,3
Babelfy / ACE2004	306	28,4	5,9	2,3	-17,3	3,6	-1,3	-18,6	3,3	-1,6
Babelfy / Ritter	1496	28,7	4,2	0,3	-15,8	0,8	-0,2	-17,9	0,6	-0,3
Babelfy / N3-Reuters-128	880	37,7	4,1	0,5	-17,5	1,7	0,5	-17,8	1,9	0,1
FRED / GERDAQ-TrainingB	433	24,0	34,2	9,7	-3,2	3,7	5,5	-15,0	14,1	-5,5
WAT / OKE 2015 Task 1 GSS	341	17,6	1,8	0,0	-12,6	3,8	0,0	-12,6	3,2	0,0
FRED / GERDAQ-TrainingA	441	24,5	33,8	12,9	0,0	0,0	0,0	-11,1	9,8	-10,2
FRED / GERDAQ-Dev	420	17,9	38,1	14,8	0,0	0,0	0,0	-10,2	9,8	-10,0
FOX / OKE 2015 Task 1 GSS	341	10,9	4,7	0,6	-9,1	2,9	-0,6	-9,1	2,6	-0,6
Babelfy / UMBC-Test	2232	12,9	3,1	0,4	-8,2	0,9	-0,3	-8,9	0,8	-0,4
AIDA / OKE 2015 Task 1 GSS	341	10	1	0	-9	3	0	-9	3	0
WAT / MSNBC	747	13,1	0,4	0,0	-6,4	5,1	0,0	-6,7	5,1	0,0
WAT / OKE 2016 Task 1 GSS	1049	12,0	1,8	0,0	-6,1	4,8	0,0	-6,1	4,6	0,0
FRED / DBpediaSpotlight	330	10,9	35,2	3,3	-2,7	0,6	-0,3	-6,1	-1,2	-3,3
Babelfy / DBpediaSpotlight	330	8,2	4,8	0,0	-5,8	1,5	0,0	-6,1	1,5	0,0
Babelfy / KORE50	144	5,6	3,5	0,0	-4,9	2,1	0,0	-4,9	1,4	0,0
Babelfy / GERDAQ-TrainingB	433	10,6	2,8	0,2	0,0	0,0	0,0	-4,8	0,2	-0,2
AIDA / MSNBC	747	5,9	0,5	0,0	-4,7	5,2	0,0	-4,7	5,2	0,0
WAT / N3-Reuters-128	880	29,9	0,0	0,0	-4,7	4,7	2,0	-4,7	4,5	2,0
WAT / Derczynski	286	15,4	0,0	0,0	-4,5	0,7	0,0	-4,5	0,7	0,0
FOX / OKE 2016 Task 1 GSS	1049	5,7	2,2	0,2	-4,4	4,0	-0,2	-4,4	3,9	-0,2
Babelfy / GERDAQ-TrainingA	441	16,6	1,4	0,2	0,0	0,0	0,0	-4,1	0,0	0,0
Babelfy / IITB	18308	5,9	3,8	0,6	-3,6	1,5	-0,5	-4,0	1,4	-0,6
Babelfy / GERDAQ-Test	409	10,5	2,0	0,0	0,0	0,0	0,0	-3,9	0,0	0,0
Babelfy / GERDAQ-Dev	420	12,1	2,6	0,2	0,0	0,0	0,0	-3,6	0,0	0,0
WAT / OKE 2015 Task 1 ED	664	9,9	1,4	0,0	-3,2	5,6	0,0	-3,2	5,6	0,0
AIDA / Derczynski	286	5,9	0,0	0,0	-3,1	0,3	0,0	-3,1	0,3	0,0
WAT / N3-RSS-500	1000	33,2	2,7	0,5	-3,0	1,4	0,2	-3,0	1,4	0,2
WAT / ACE2004	306	15,7	2,6	0,0	-2,9	5,2	0,3	-2,9	5,2	0,3
FOX / Derczynski	286	6,3	2,1	0,0	-2,4	0,3	0,0	-2,4	0,3	0,0
WAT / OKE 2016 Task 1 ED	340	5,6	0,3	0,0	-2,4	5,3	0,0	-2,4	5,3	0,0
FOX / N3-Reuters-128	880	6,5	8,0	0,1	-1,7	4,7	0,0	-1,7	4,5	0,0
Legend: GSS = Gold Standard Sample ED = Evaluation Dataset				Average:	-8,7	2,3	0,1	-11,7	3,1	-1,3

Figure 6 shows the F-measure gains obtained by applying MInT NoIn and MInT NoOver to enhance mentions recognized by Babelfy, the annotator with the highest number of mentions corrected by using MInT. Notice that MInT improved the

performance in most of the datasets. In addition, MInT NoIn has provided superior gains than MInT NoOver in most cases.

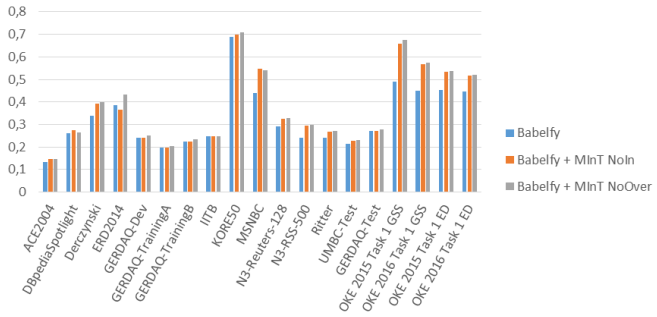


Figure 6: F-Measure gains of MInT over Babelfy

Finally, Figure 7 shows the increasing or decreasing of micro F1-measures that the different annotators achieved on the datasets when using MInT NoIn and MInT NoOver. It can be seen that the effect of both approaches highly depend on the used NER system and the dataset. While the scores of Babelfy are nearly always increasing when using MInT NoIn or MInT NoOver, the scores of the other systems are more ambiguous and depend on the dataset. The scores of Babelfy are nearly always increasing when using MInt NoInt or MInt NoOver. This occurs because over segmentation is a problem for Babelfy, which returns segmented mentions such as “U.S.”, “U.S. National” and “National Institutes”. The MInT algorithms merge such mentions and, in this example, produces the correct mention “U.S. National Institutes of Health”.

The scores of the other systems are more ambiguous and depend on the dataset. The F1-scores for ERD2014 and GERDAQ are nearly not influenced. Since both contain only short queries without a sentence structure, three of the tools return no mention at all, which can not be improved by any MInT algorithm. Additionally, the creators of the GERDAQ datasets tended to separate long named entities into shorter named entities, e.g., the query "fort desoto fishig report" is annotated as “fort desoto”, “fishig” and “report”, which contradicts the aim of MInT. This leads to a decreasing of the performance of the FRED system on the GERDAQ datasets when combined with MInT, while the same system increases its performance when tested with the ERD2014 dataset. The latter dataset contains longer mentions and MInT merges the short mentions the FRED system generated. Derczynski as well as the OKE 2015 Task 1 gold standard sample are datasets for which the F1-score for all systems is highly increased when using MInT. On the Derczynski dataset, MInT helps to extend the range of named entities that the NER systems are able to identify, e.g., they replace the two mentions “Miami” and “Ibiza” by the correct song name “Miami 2 Ibiza”. The OKE 2015 dataset, comprises documents with many locations from the US that are written like "Muscatine, Iowa". While the most NER systems identify this as two separated entities, MInT merges them which aligns with the guidelines of the OKE 2015. The difference to the performance of the MInT approaches on the OKE 2015 Task 1 evaluation dataset is caused by a topic shift. The latter mainly comprises documents describing locations outside of the US, where this writing is not common. ACE2004 is a dataset for which the system performance is decreased when using MInT. This is partly caused by location names like “Davenport, Iowa”, which is combined by the MInT approaches as explained above while the gold standard only expects “Iowa”. However, based on [20] the results on this dataset shouldn’t be weighted to high and are listed here only for completeness.

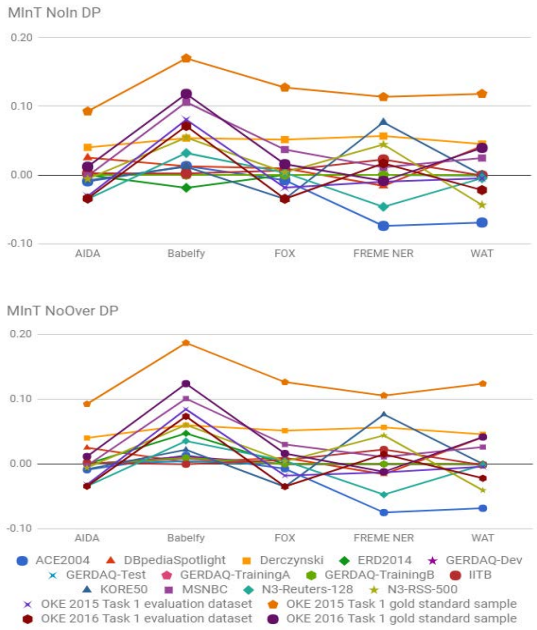


Figure 7: F1-measure gains of MInT with DP surface names

5.3 Discussion

The major findings of the experiments can be summarized as: (i) there is a higher prevalence of over-segmentation than other partial matching problems in the mentions recognized by the vast majority of the annotators in the majority of the datasets considered in the experiments; (ii) MInT can solve most cases of over-segmentation for many annotators and datasets; (iii) MInT causes under-segmentation as a side effect, with a higher intensity when using surface names that are not in the ground truth; (iv) MInT can improve F-Measure, with a higher intensity with an ideal dictionary (surface names of the ground truth); (v) MInT NoOver is prone to cause more cases of under-segmentation as a side-effect than MInT NoIN. Overall, both approaches MInT NoIn and MInT NoOver have the ability to increase the performance of NER systems by up to 0.19 F1-score. On average, MInT NoOver increases the F1-score more than MInT NoIn. However, the results clearly show that the usage of these approaches are bound to the NER system, the type of text data and whether the creation of long mentions is intended.

6 CONCLUSIONS

The classification of incompatibility problems (including partial matches) between mentions identified in text and mentions in a ground truth contributes to a better diagnosis of the situation than

by using just performance measures such as precision and recall. The analysis of the HAREM CD-2 corpus in Portuguese with the annotator Priberam REM, and a variety of prominent corpuses applied to several state-of-the-art annotators currently available in Gerbil revealed a greater prevalence of over-segmentation than other mention partial matches. It suggests that mention expansion can contribute to improve mention recognition performance. Thus, the MInT algorithms were developed for solving over-segmentation. They rely on some dictionary of surface names for doing mention expansion. One advantage of our dictionary-based approach for mention expansion is that it does not impose restrictions on surface names. It allows surface names containing special characters and punctuation signs, what usually have a negative impact in the performance of mention recognition tools. The MInT algorithms differ in the way they cope with mention overlapping caused by mention expansion to surrounding text, what results in slightly different performance.

The experimental results presented in this paper support the following conclusions: (i) over-segmentation is, by far in some cases, the most prevalent of the partial matching problems in a variety of datasets; (ii) MInT can correct most cases of over-segmentation without causing relevant side-effects, at least with ideal dictionaries (derived from the mentions in the ground truth); and (iii) F-measure of mention recognition can be improved by employing MInT. In addition, we have observed, in various cases (e.g., examples in table 1 and examples 3.1, 3.2 e 3.3) and experimental results, that longer mentions are frequently associated with more specific information (e.g., *US Defense Department* refers to something more specific than *US*).

Future work includes: (i) performing experiments to evaluate the performance of MInT using a wider variety of languages and dictionaries of surface names available for real world application; (ii) developing methods for mention enhancing that do not depend on dictionaries to drive mention expansion; (iii) investigating methods that also help to correctly classify mentions after their correction; (iv) better investigate the correlation between mention size, in terms of number of terms, and how specific the mention is; and (v) applying the proposed method to fields such as social media analytics, in which the (correct) analysis of social contents is of high interest, but still suffers shortcomings caused by difficulties to segment and disambiguate mentions [21]. Over-segmentation could mean information-loss in fields such as of social media analysis, in which smaller text segments imply a reduction in the already limited context available to disambiguate mentions. Furthermore, improved mention segmentation is crucial to increase the realiabiliy of text analysis in social media.

ACKNOWLEDGMENTS

This work was partially supported by CAPES (grant 88881.121467/2016-01), DAAD, EFRE Fund European Union project S2DES, the H2020 project HOBBIT (GA no. 688227) as well as the EuroStars projects DIESEL (project no. 01QE1512C) and QAMEL (project no. 01QE1549C).

REFERENCES

[1] Ratinov, L., Roth, D. (2009, June). Design challenges and misconceptions in named entity recognition. In 13th Conf. on Computational Natural Language Learning (CoNLL '09). Association for Computational Linguistics Stroudsburg, PA, USA, 147-155.

[2] Ratinov, L., Roth, D., Downey, D., & Anderson, M. (2011, June). Local and global algorithms for disambiguation to wikipedia. In 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT '11), Stroudsburg, PA, USA, 1375-1384.

[3] Avirup Sil and Alexander Yates. 2013. Re-ranking for joint named-entity recognition and linking. In 22nd ACM Intl. Conf. on Information & Knowledge Management (CIKM '13). New York, NY, USA, 2369-2374.

[4] Luo, G., Huang, X., Lin, C. Y., & Nie, Z. (2015, September). Joint named entity recognition and disambiguation. In Conf. on Empirical Methods in Natural Language Processing (EMNLP '15). Association for Computational Linguistics Lisboa, Portugal, 879-888.

[5] Nguyen, D. B., Theobald, M., & Weikum, G. (2016). J-NERD: joint named entity recognition and disambiguation with rich linguistic features. Transactions of the Association for Computational Linguistics (TACL 4). Association for Computational Linguistics, 4, 215-229.

[6] Mihalcea, R., & Csomai, A. (2007, November). Wikify!: linking documents to encyclopedic knowledge. In 16th ACM Conf. on Information and Knowledge Management (CIKM '07). ACM New York, NY, USA, 233-242. DOI= https://doi.org/10.1145/1321440.1321475

[7] Mendes, P. N., Jakob, M., García-Silva, A., & Bizer, C. (2011, September). DBpedia spotlight: shedding light on the web of documents. Proceedings of the 7th International Conference on Semantic Systems (I-Semantics '11). ACM New York, NY, USA, 1-8.

[8] Chiu, Y. P., Shih, Y. S., Lee, Y. Y., Shao, C. C., Cai, M. L., Wei, S. L., & Chen, H. H. (2014, July). NTUNLP approaches to recognizing and disambiguating entities in long and short text at the ERD challenge 2014. In Proce. of the 1st Intl. workshop on Entity recognition & disambiguation (ERD '14). ACM New York, NY, USA, 3-12.

[9] Gamallo, P., & Garcia, M. (2011, October). A resource-based method for named entity extraction and classification. In Proceedings of the 15th Portuguese Conference on Artificial Intelligence (EPIA 2011), Springer Berlin Heidelberg, Lisboa, Portugal, 610-623.

[10] Li, G., Deng, D., & Feng, J. (2010, October). Extending dictionary-based entity extraction to tolerate errors. In 19th ACM Intl. Conf. on Information and knowledge management (CIKM). New York, NY, USA, 1341-1344.

[11] Deng, D., Li, G., Feng, J., Duan, Y., & Gong, Z. (2015). A unified framework for approximate dictionary-based entity extraction. The VLDB Journal Springer Inc. Secaucus, NJ, USA, 24(1), 143-167.

[12] Plu, J., Rizzo, G., & Troncy, R. (2015, May). A hybrid approach for entity recognition and linking. In Semantic Web Evaluation Challenge (ESWC '15). Springer Intl. Publishing Switzerland Portoroz, Slovenia, 28-39.

[13] Speck, R., & Ngomo, A. C. N. (2014, October). Ensemble learning for named entity recognition. In 13th Intl. Semantic Web Conference – Part I (ISWC 2014). Springer International Publishing Switzerland, 519-534.

[14] Usbeck, R., Ngomo, A. N., R¨oder, M., Gerber, D., Coelho, S. A., Auer, S., and Both, A. (2014). AGDISTIS- graph-based disambiguation of named entities using linked data. In 13th Intl. Semantic Web Conf. (ISWC 2014). Springer Intl. Publishing Switzerland, LNCS, 8796, 457–471.

[15] Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. (2013). A framework for benchmarking entity-annotation systems. In 22nd Intl. Conf. on World Wide Web (WWW '13). ACM, New York, NY, USA, 249-260.

[16] Michael Röder, Ricardo Usbeck and Axel-Cyrille Ngonga Ngomo. 2017. GERBIL – Benchmarking Named Entity Recognition and Linking Consistently. In 24th Intl. Conf. on World Wide Web (WWW '15). Republic and Canton of Geneva, Switzerland, 1133-1143.

[17] Amaral, C., Figueira, H., Mendes, A., Afonso, A.D. (2008). Adaptation of the Priberam entity recognition system to HAREM. In: MOTA, C and SANTOS, D. Challenges in the evaluation of named entity recognition: The Second HAREM. Linguateca.

[18] Moro, A., Raganato, A., Navigli R. Entity (2014) Linking meets Word Sense Disambiguation: a Unified Approach. Transactions of the Association for Computational Linguistics (TACL), 2, pp. 231-244.

[19] Gangemi, A., Presutti, V., Recupero, D., R., Nuzzolese, A., G., Draicchio, F., Mongiovì, M. (2016) Semantic Web Machine Reading with FRED. Semantic Web Journal (preprint version).

[20] Jha, K., Röder, M., Ngomo, A. C. N. (2017) All That Glitters is not Gold -- Rule-Based Curation of Reference Datasets for Named Entity Recognition and Entity Linking. In: 14th Intl. Conf. on The Semantic Web. Latest Advances and New Domains (ESWC 2017).

[21] Alt, R., Wittwer, M. (2014). Towards an Ontology-based Approach for Social Media Analysis. In 22nd European Conf. on Inf. Systems. pp. 1–10.