Preprocessing Does Matter: Parsing Non-Segmented Arabic

Noor Abo Mokh, Sandra Kübler
Indiana University, Bloomington, IN
noorabom@iu.edu, skuebler@indiana.edu

ABSTRACT
Preprocessing is a normal first step in parsing, but it is the step that most researchers consider trivial and not worth reporting. The problem is exacerbated by the fact that parsing research often focuses on parsing a treebank rather than parsing a text since the treebank obscures many of the preprocessing steps that have gone into the curation of the text. In this paper, we argue that preprocessing has a non-negligible effect on parsing, and that we need to be careful in documenting our preprocessing steps in order to ensure replicability. We focus on parsing Arabic since Arabic is more difficult than English in the sense that 1) the orthography has intricacies such as vocalization that need to be handled and that 2) the basic units in the treebank do not necessarily correspond to words but sometimes constitute morphemes. The latter necessitates the use of a segmenter in order to convert the text to a form that the parser has seen in training. We investigate a scenario where we combine a morphological analyzer/segmenter, MADAMIRA, with a parser trained on the Arabic Treebank. We mainly examine the differences in orthographic and segmentation decisions between the analyzer and the treebank. We show that normalizing the two representations is not a simple process and that results can be artificially low or misleading if we do not pay attention. In other words, this paper is an attempt at establishing best practices for parsing Arabic, but also more generally for documenting preprocessing more carefully.

KEYWORDS: Parsing, Arabic, preprocessing.

1 Introduction

Preprocessing is a normal step before parsing, it encompasses all the steps to convert the text to be parsed so as to model the decisions in the treebank that was used to train the parser. Details of preprocessing are generally not reported in the parsing literature because this step is considered trivial and not part of the research. Also, in parsing Arabic literature, gold segmentation of the text to be parsed is usually assumed. This means that that we ignore the issues of converting the text to be parsed to the format seen in traning the parser. However, the decisions taken in preprocessing are of vital importance to ensure parsability and can have major effects on parsing if we are not careful.

Our focus is on parsing Arabic, and our ultimate goal is research on integrating segmentation decisions jointly with the parsing step by presenting the different possible segmentations for words in a lattice to the parser. Such a setting is more representative of naturally occurring data, as no gold segmentation is assumed.

However, in working with the state of the art morphological analyzer/segmenter for Arabic, MADAMIRA (Pasha et al., 2014), we found that there are differences in segmentation decisions and orthographic normalizations between the treebank and the segmenter (for details see section 2) . These need to be addressed before we can successfully combine the segmenter and the parser. This paper describes the non-trivial process of finding a common representation. If these issues are not addressed carefully, parsing results will be artificially low or severely misleading.

All of the issues addressed in this work are considered preprocessing, and to our knowledge are not described in publications on Arabic parsing in enough detail to allow for replicability. Thus this paper is intended as an attempt at establishing best practices for parsing Arabic, but also more generally for documenting preprocessing carefully enough to ensure replicability. Note that these issues do not occur when we focus on parsing treebank data since we then implicitly use the same (potentially undocumented) preprocessing used in the treebank.

The remainder of this paper is organized as follows: Section 2 presents a closer inspection of the issues that we address in the following sections. In section 3, we show how preprocessing has been handled and documented in previous work, or how published work is often short on details. In section 4, the experimental design is described, including a description of the treebank and tools used in this paper. Section 5 explains the process of adjusting MADAMIRA analyses to mirror the treebank decisions and shows the effects that the individual steps have on parsability. We conclude by a discussion of the bigger picture of preprocessing for parsing Arabic in section 6.

2 Preprocessing Issues

When we envision the use of a parser in a realistic scenario, the parser needs to handle standard text, split into sentences. However, parsing research generally focuses on parsing data from treebanks, which has already been preprocessed. For English, the discrepancy between 'realistic' text and treebank text is not dramatic, it mostly consists of splitting contractions such as "you'll" or "can't".

In languages such as Arabic, the difference is much greater, for a range of reasons. One of the differences concerns the fact that in Arabic, clitics such as pronouns and prepositions are agglutinated as affixes to the subsequent word. In the treebanks, these clitics tend to be split

off and treated as separate words, thus necessitating a segmentation step before we can parse new text. However, the problem is complicated by the fact that the segmentation of a surface word may be context dependent and can thus only be reliably performed during parsing. For example, the word الَام "AlAm" [1] can be segmented in three different ways, with very different meanings and morpho-syntactic structure: الـ أم "Al >m" (Eng.: the mother), آلَام "|lAm" (Eng.: pains), أ لَام "> lAm" (Eng.: did he blame?). Another issue, which is intricately connected to the segmentation step, concerns orthographic normalization, for example, vocalization and Hamza normalization. Arabic is normally written without short vowels and some Hamza variants, but the treebank may contain those because they often disambiguate the word. Also, the different spelling variants, e.g., of the Hamza, are sometimes inconsistently used by Arabic users (where some writers may ignore the Hamzas or use a different Hamza variant). Segmenters and treebanks may decide to normalize to one Hamza variant, but not necessarily to the same one.

The sentence in (1) shows an example of the variation between original text, MADAMIRA (Pasha et al., 2014) analyses, and the representation in the Penn Arabic Treebank[2] (Seddah et al., 2013), with all the phenomena discussed above. The source text is not segmented, MADAMIRA provides automatic segmentation, and the Arabic Treebank provides gold segmentation.

(1) a. Arabic script:

الجغرَافي لتُؤكد نظريتهَا

Eng.: 'the geographic to emphasize her theory'

b. Source text: **AljgrAfy** lt&kd **nZrythA**
c. Treebank: Al**jgrAfy** l t&kd **nZryt hA**
d. MADAMIRA: Al **jgrAfy** l t&kd **nZryp hA**

When we compare the three versions, we expect the difference in segmentation between the source text on the one hand and MADAMIRA and treebank on the other hand. However, we also see additional differences: First, the treebank and MADAMIRA make different segmentation decisions: While MADAMIRA splits of the definite article "Al" in الجغرَافي "AljgrAphy" (Eng.: geographic), the treebank does not. Second, the feminine marker representation in the treebank (نظريتـ, "nZryt", Eng.: theory) does not match with the representation in MADAMIRA (نظرية, "nZryp").

To the best of our knowledge, these decisions are not documented fully, but they need to be handled in preprocessing if we want to successfully parse text that was segmented by MADAMIRA.

3 Related Work

3.1 Preprocessing

Preprocessing can concern issues of orthographic normalization, segmentation, the data splits, and converting trees into training data. While we are only concerned with the first two issues, all such steps need to be reported: Dakota and Kübler (2017) have shown that decisions in

[1] We use Buckwalter transliteration (Buckwalter, 2004) throughout the paper since this is also used in the treebank and segmentation step.

[2] In the version used in the SPMRL shared tasks 2013/2014.

preprocessing can have a significant impact on parsing results. A study by Goodman (1996) shows that he was not able to duplicate the results of Bod (1996) because Bod did not give sufficient details on either the preprocessing of the data or the data split. Also, Cheung and Penn (2009) report that they were not able to replicate experiments by Becker and Frank (2002) because Becker and Frank performed manual correction which were not documented.

Given this situation, a detailed description of all preprocessing steps is indispensable to enable replicability of such studies.

There are only very few parsing papers that broached the topic explicitly: Wagner et al. (2007) detail the preprocessing steps performed on the data from the BNC to match the Penn Treebank representation. Seddah et al. (2012) describe preprocessing steps such as normalization, style modifications, spell corrections, for adaptation of noisy user-generated content to a Penn treebank-based parser.

However, in much of the current parsing literature, preprocessing is minimally described, if it is mentioned at all. The lack of preprocessing information can be found in parsing studies for many languages. For English, Bod (2001) describes that all trees were stripped off their semantic tags, coreference information, and quotation marks but no other details are mentioned. Charniak and Johnson (2005) mention that they used "the division into preliminary training and preliminary development data sets described in (Collins and Koo, 2005)" but no other preprocessing details are provided. In (Collins and Koo, 2005), only information on the data split is mentioned and no other preprocessing steps are documented, this is also the case for work by McClosky et al. (2006). Klein and Manning (2003) mentions the tree annotation and transformation were similar to the ones described by Johnson (1998). While Johnson (1998) details his method of tree transformation, only the data splits are mentioned. Petrov and Klein (2007), work on English, Chinese and German, they do not provide information about preprocessing.

3.2 Preprocessing Arabic

Preprocessing steps in parsing Arabic studies is also minimally described which makes it difficult to duplicate results of such studies. Most of the studies on Arabic parsing, specifically in parsing of Modern Standard Arabic (MSA) studies, use treebank data where gold segmentation is usually assumed. Preprocessing of such text is minimally described, as it is often presumed that the data sets are similar, i.e., they use matching segmentation schemes and preprocessing steps in the training data and in the test data.

In their work on dialectal Arabic, Chiang et al. (2006) describe the data split in addition to the tree transformations, they also mention that they use the undiacriticized form. Al-Emran et al. (2015),i n a study on parsing MSA, also used treebank data, but no other details on preprocessing are mentioned. In (Attia et al., 2010), only data splits are mentioned, but no details about the preprocessing steps. In the SPMRL shared task description (Seddah et al., 2013, 2014), the specifics of normalization are described: "all tokens are minimally normalized: no diacritics, no normalization except for the minimal normalization present in MADA's back-end tools". It is also mentioned that they follow the standard ATB segmentation, where categories of orthographic clitics are split except from the article Al+.

Other studies use an external tool for segmentation and tokenization. Kulick and Bies (2009) used a morphological analyzer, but no preprocessing steps were documented, only the data split is mentioned. However, they mention that since there is an issue of mismatching tokens, they

did not report a parsing score, because of the evaluation method. I.e., the tokens do not match because of differences in segmentation schemes used, hence, the constituent spans cannot be compared.

Green and Manning (2010) experiment with two scenario, one where gold segmentation is assumed, and one where a joint segmentation and parsing experiment is performed. In the gold scenario, preprocessing steps mentioned are removing all diacritics, normalizing Alif variants, mapping the Arabic punctuation marks to their Latin equivalents, and segmentation markers were kept. They used this normalization because other orthographic normalization schemes suggested by Habash and Sadat (2006) which they experimented with, had an insignificant influence on parsing performance in comparison to their normalization scheme. In their non-gold scenario, they experiment with two different pipelines, one where they use a manually-created list of clitics provided by Maamouri et al. (2004), to generate the lattices. They mention that no other preprocessing was made for this setting besides a correction rule for a deleted 'A' from a determiner 'Al'. This pipeline was compared to another pipeline in which Green and Manning (2010) used MADA (v3.0) to generate 1-best analyses. No other information or details are mentioned about preprocessing.

Another question which often remains unclear concerns the issue that in naturally occurring data, some diacritics or Hamza variants might be present in the text to be parsed. Therefore, it is unclear how much normalization is needed. This was also addressed by Maamouri et al. (2008), who show that while non-diacritized forms are the default, it might not be representative of real world data.

4 Experimental Setup

4.1 Treebank

For our experiments, we use the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) from the 2013 SPMRL Shared Task (Seddah et al., 2013, 2014). The shared tasks provided constituent trees and dependency trees, the latter based on the annotations of the Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009). The two versions are aligned at the word level.

For our experiments, we use the constituent version. The training data we use are unvocalized, except from few cases of Hamzas (considering that Hamza is a form of diacritization).

For training, we use the 5k dataset (i.e., the first 5 000 sentences of the complete training file), and we use the dedicated test set containing 1 959 sentences.

4.2 MADAMIRA

To segment the original text, we use the morphological analyzer MADAMIRA (Pasha et al., 2014). MADAMIRA is a morphological analyzer for Arabic (Pasha et al., 2014), a combination of MADA (Habash et al., 2009; Habash and Rambow, 2005) and AMIRA (Diab et al., 2004; Diab, 2009), which performs segmentation as part of the morphological analysis.

Text in Arabic script is transliterated into Buckwalter transliteration (Buckwalter, 2004). The morphological analyzer produces a set of possible analyses for each word. This list is then processed by the feature modeling component, where an SVM and language models are applied to create predictions of morphological features and other features such as diacritics for each token. The list is ranked based on the model predictions, and the text is tokenized based on morphological features. For our current work, we use only the highest ranked analyses.

4.3 Parser

We use the *Blatt Parser* (Goldberg and Elhadad, 2011), a reimplementaiton of the Berkeley Parser (Petrov et al., 2006; Petrov and Klein, 2007) which was extended to allow lattices as input. We use five split-merge cycles for training.

4.4 Evaluation

For the evaluation, we use the scorer from the SPMRL 2013 Shared Task (Seddah et al., 2013). The scorer is derived from the *EVALB* version used for the SANCL 2012 "Parsing the Web" shared task (Petrov and McDonald, 2012), which in turn is based on teh version by Sekine and Collins (1997); with additional options, such as allowing the evaluation of function labels and penalizing unparsed sentences. EVALB, in contrast, reports the number of unparsed sentences. However, they are ignored in the calculation of precision and recall.

For the purposes of the work reported here, we are not interested in the standard evaluation metrics. Instead, we focus on the errors report, i.e., the cases flagged by the evaluation script as containing errors. Such errors can consist of differences in the words in the parsed text and the treebank, rather than in the tree structures. The number of mismatches serves as an indicator of how different the MADAMIRA analyses are from the treebank sentences. We do not use the standard evaluation metrics because we are interested in the differences in spelling and segmentation between the parser output and treebank sentences, which can be found in the errors report. Looking at the parsing results may not be representative of the actual results if many sentences are ignored because of such differences. For instance, assuming only 100 sentences were parsed without errors out of the 1 959 sentences, and the score is 97.2, then this is inaccurate as the score does not reflect the errors found in sentences.

In the errors report, there are two types of error messages; one flags sentences where there is a mismatch in sentence length, i.e., the number of tokens in the sentence segmented by MADAMIRA does not match the number of tokens in the treebank. Mismatch in length can be as a result of segmentation differences. For instance, the phrase الجولة السَّادسة "Aljwlp AlsAdsp" (Eng.: the sixth round), from the treebank is analyzed as "Al jwlp Al sAdsp". Splitting off the determiner 'Al' is not an error, but is a case where MADAMIRA and the treebank made different decisions.

The second type of errors flags tokens which do not match with their corresponding tokens in the treebank. We see this type of errors occurring because of differences in orthographic normalization, for example in Hamza representation: MADAMIRA returns إنتظر "{ntZr" (Eng.: waited), which is represented as انتظر "AntZr" in the treebank; i.e., the treebank normalizes the Hamza { to A.

Note that there is an interdependence between the two error types since token mismatches are only reported if the number of tokens matches between parser output and treebank. This means that when we correct a segmentation difference, such as the split-off determiner described above, the number of length mismatches will decrease, but the number of token mismatches may increase since the corrected sentences are now checked for token mismatches.

5 Preprocessing Steps

As described above, our ultimate goal is to create different segmentation hypotheses using MADAMIRA and then make final segmentation decisions during parsing. Our current work

	Normalization	# length errors	# token errors/diff.	# no error	# skipped sent.
1	Baseline	1 899	50	3	7
2	+ delete diacritics	1 900	16	37	6
3	+ fixed determiner	582	836	541	0
4	+ modified fem. pronouns	582	795	582	0
5	+ modified numbers	469	865	625	0

Table 1: First steps of normalization and their effect on parsability.

focuses on the preprocessing steps necessary to create an interface between MADAMIRA and the Arabic Treebank representations. This means that we need to determine the differences in representations and normalize them automatically as far as possible. We assume that there will be some differences that are too idiosyncratic in nature to be handled automatically and would need manual intervention. We use the SPMRL scorer to find all differences. However, the procedure is complicated by the fact that some of the differences will result from incorrect segmentation. These latter differences concern our ultimate goal and will not be addressed here since we need to make sure that we handle all the preprocessing decisions before we address the problem of segmentation.

In the following sections, we describe the baseline and all the modifications along with their effect on the evaluation. An overview of these effects can be found in table 1.

5.1 Baseline Experiment

The baseline experiment simulates the case where we present MADAMIRA with standard text (in Buckwalter transliteration). We use MADAMIRA's analyses as input for the parser, with only one modification: The standard text from the treebank contains the symbols -LBR- and -RBR- instead of opening and closing parentheses, which we changed back to the original parentheses. The reason why the parentheses in the text were replaced is that many parsers cannot distinguish between parentheses as part of the sentence and syntactic brackets, which are also marked by parentheses. MADAMIRA, in contrast, expects parentheses, not the replacement symbols.

MADAMIRA provides for each word its diacriticized form, part of speech, Buckwalter transliteration, the lemma, and morphological information. For the purpose of this experiment, segments of a diacriticizised token were extracted. For instance, لمُؤتمر "lm&tmr" (Eng.: to a conference), is segmented into 'li' ل and 'mu&otamari', where 'li' is a preposition, and 'mu&otamari' is a noun. Diacritics were kept in the representation of each token for the baseline experiment since ultimately we need to extract all possible segmentations, which are only available in the diacriticized form from MADAMIRA. Based on this experiment, only three sentences did not cause any errors in the evaluation script. I.e., all other sentences could not be evaluated because of mismatches on the word level. This was not unexpected, as diacritics are not included in the treebank[3]. However, this result makes it very obvious that we need to adapt MADAMIRA analyses before parsing.

5.2 Deleting Diacritics

The obvious next step is to delete diacritics. No other modifications are performed on the data. Based on this setting, the scorer reports 1 916 errors, 1 900 of which concern sentences with

[3]Here we refer to the dataset used in this study, other releases of the treebank data include diacritized forms

length mismatches (see line 2 in table 1). The number of sentences without errors increases from 3 to 40, which is a much lower improvement than expected, indicating that there are additional, severe problems in the sentences.

5.3 Handling the Definite Article 'Al'

Since removing the diacritics did not improve results significantly, the next step is to determine other causes for length and thus represenetational mismatches. Looking through the sentences that were flagged for length mismatch, we found an issue concerning the segmentation of the determiner 'Al' in noun phrases. In written forms in Arabic, the determiner is usually connected to the noun or adjective, forming one orthographic unit. But in MADAMIRA's analyses, the determiner and noun or adjective are segmented into two different tokens, as described in section 4.4. This is different from the treebank, where the determiner and noun or adjective occur as one orthographic unit. For instance, the Arabic word التّفّاحة "AltfAHp" (Eng.: apple) is represented as two segments in MADAMIRA, 'Al' ال, the determiner, and 'tfAHp', the noun. In the treebank, this word is represented as "Al+tfAHp", with determiner and noun forming a single token. This difference results in a length mismatch in the evaluation script.

After reattaching the determiner, the number of sentences with length mismatches decreases dramatically from 1 900 to 582, and the number of sentences without errors increases from 40 to 541 (see line 3 in table 1). However, the number of token mismatches increased considerably as well, to 836. This means that many of the sentences that do not have a length mismatch anymore still have a token mismatch and thus cannot be evaluated.

5.4 Feminine Marker Ta-Marbuta

Another look at the data shows that out of the 836 token mismatch cases, there are 138 sentences involving the feminine marker Ta-Marbuta. Overall, there are over 300 cases with a mismatch of the Ta-Marbuta in those sentences. The feminine marker Ta-Marbuta 'p' is transliterated as 't' in the treebank if it is followed by a pronominal clitic. In MADAMIRA's analyses, however, the modification of the Ta-Marbuta remains 'p', even if followed by a clitic. For instance, كتابته "ktAbth" (Eng.: his writing) is segmented by MADAMIRA as كتابة ه "ktAbp h" while it is represented as كتابتـ ه "ktAbt h" in the treebank.

Therefore, we change the transliteration of the feminine marker when it is followed by a pronoun in MADAMIRA's analyses to match the representation in the treebank. The evaluation results show that the number of length mismatches does not change. This is to be expected since the Ta-Marbuta normalization does not change the segmentation, but only the word-internal representation. The number of token mismatches decreases slightly, from 836 to 795 (see line 4 in table 1). In addition, the number of sentences without errors increases from 541 to 582. It is worth noting that the decrease in token mismatches does not match the number of sentences displaying this issue. This means that some of the sentences in which the Ta-Marbuta was normalized still contain other mismatches.

5.5 Numbers representation

In the next step, when checking the texts again, we found another frequent difference in the representation of numbers separated by a hyphen. As an example, in the treebank, hyphenated

Hamza Type	# in MADAMIRA	# in Treebank
lone Hamza (')	1 017	1 017
Hamza on Wa (&)	417	417
Hamza on Ya (})	1 303	1 303
Hamza above Alif (>)	6 140	2 156
Hamza below Alif (<)	2 683	917
Alif Al-Wasla ({)	1 577	0
Madda above Alif (|)	217	173
Alif Maqsura (Y)	2 379	2 447

Table 2: Hamza types in MADAMIRA and the Penn Arabic Treebank.

numbers are represented as '2-1'; however, in MADAMIRA, these were segmented into '2 - 1'.

After modifying the number representations, the number of length mismatches decreases (see line 5 in table 1). This is expected, given the fact that the former segmentation of numbers affect the length of sentences, hence resulting in such errors.

5.6 Hamza and Alif Inconsistencies

Besides the mismatches described above, there are issues involving the Hamza and Alif representation. Alif is a letter while Hamza is a diacritic-like letter mark (Habash, 2010). I.e., the Hamza has similarities with a diacritic, but Habash notes that "The general consensus on encoding the Hamza is to consider it a letter mark (and as such part of the letter) as opposed to being a diacritic". The Hamza can appear on letters such as Alif, Wa and Ya. The form of the Hamza varies depending on its position in the word.

Hamza inconsistencies can be due to inconsistencies in spelling in Arabic texts rather than normalization efforts during segmentation and analysis since Arabic writers often do not write the Hamza or use different variants of Hamza, thus making it an optional mark. This is the reason why variants of Hamza that occur with the letter Alif are usually normalized.

Examining the segmentation in MADAMIRA and the treebank text, there seem to be differences in the spelling choices. Table 2 shows the frequencies of Hamza and Alif variants in both variants. While some Hamzas (lone, on Wa, and on Ya) match in numbers in the treebank and in MADAMIRA, it seems that <, > and { (an alif variant) cause most of the inconsistencies, as indicated by the differences in counts between the two representations.

Due to the inconsistencies in Hamza and Alif spelling, we investigate two approaches: We first modify the Hamza representation in the MADAMIRA analyses such that specific variants of Hamzas are changed to match the treebank representations. In case the modification cannot be performed automatically, we use the second approach, which is a more traditional approach, where Hamza variants are normalized to 'A' in MADAMIRA in one experiment, and in both MADAMIRA and treebank in the other experiment.

In the first approach, we extract all tokens where Hamza occurs and compare them to the gold dataset representation. Based on the counts of Hamzas in table 2, the most serious discrepancy concerns the Alif Al Wasla '{' because it is only used in MADAMIRA but not in the treebank. Since it is not clear what the best normalization for the Alif Al Wasla to approximate the treebank representation of the concerned words, we perform three different experiments where only

Normalization	# length errors	# token errros	# no error
result from table 1	469	865	625
Alif Al-Wasla → Hamza below Alif	469	865	625
Alif Al-Wasla → Hamza above Alif	469	862	628
Alif Al-Wasla → A	469	391	1099
Hamza above Alif → A	469	1 314	176
Hamza above Alif → Hamza below Alif	469	1 309	181
Normalized Hamza in MADAMIRA	469	1 352	138
+ in treebank	469	229	1261

Table 3: Hamza Modification following two different approaches.

specific variants of Hamza are are used, to see which nromalization would give us the closest match to the treebank representation.

The Alif Al-Wasla '{' representation was changed to three different possible normalized forms: 1) Hamza below Alif <, 2) Hamza above Alif >, and 3) bare Alif 'A'. Additionally, we investigated whether the Hamza above Alif should be normalized to a bare Alif 'A' or to Hamza below Alif '<'.

The second approach, which is often the traditional way of addressing Hamza spelling inconsistencies, is normalization of Hamza variants to bare 'A'. The Hamzas that are normalized are: lone Hamza, Madda on Alif, Hamza above and below Alif, Alif Al-Wasla. Hamza on Wa and Ya are not normalized, because their counts are similar in both datasets (as shown in Table 3). Also, normally such Hamzas are kept in writing. In one experiment, Hamza variants were normalized in the training set and in MADAMIRA analyses. The second experiment included also normalization of Hamza variants in the treebank.

The results are shown in table 3. The best results of the first method were achieved when changing Alif Al-Wasla '{' to bare Alif 'A'. This is for two reasons: First, '{' is not represented in the treebank. Second, in many cases where '{' is used in MADAMIRA, 'A' is used in treebank.

Trying to normalize the Hamza above Alif either to A or to the Hamza below Alif increases the number of token errors, i.e., is not a feasible normalization. For the second set of experiments, where all Hamza instances were normalized (excluding Hamza on Wa or Ya), we found that we need to normalize the Hamza in both MADAMIRA analyses and in the treebank in order to obtain a good normalization. This normalization results in the highest number of sentences without errors, 1 261 out of 1 959. However, note that this is a somewhat radical step since it involves modifying the gold standard.

5.7 Remaining Issues

The results in Table 3 show that we can reduce the number of both types of errors/differences considerably. However, we did not manage to reduce the number of length or tokens mismatches in all cases. This was expected: The remaining errors are a result of segmentation differences. Most of these segmentation differences cannot be solved by normalization or simple modifications. Instead, they require syntactic analysis in combination with morphological analysis. For instance, the prepositional phrase لَّاجىء "llAjY" (Eng. to a refugee, or to the refugee) was represented as لَاجىء ال ل "l Al lAjY" (Eng.: to the refugee) in MADAMIRA, while based on the gold segmentation it is "l lAj}" (ENG: to a refugee) (note the difference between the definite

and indefinite forms). In this case, more information is needed to determine the correct form (whether it is an iDafa construction or a noun-adjective construction).

Another issue we also encountered concerns the Alif Maqsura (Y), a variant of Alif, whose spelling changes if followed by an affix to either 'y' or 'A'. The problem we encountered in this case was that a word like علَى "ElY" (Eng.: on) is represented as علِيّ "Ely" in MADAMIRA, but as "ElY" in the treebank if followed by a suffix. One way to overcome this issue is normalizing all instances of Alif Maqsura. However, this will create more errors since there were other instances where Alif Maqsura 'Y' is used for adjectives in the treebank, but MADAMIRA would analyze them as nisba adjective ending (i.e., -y). Normalizing all instances of 'Y' would result in errors for instances were the correct analysis is predicted by MADAMIRA. For this reason, we did not normalize the Alif Maqsura.

There are additional differences that need manual modifications, especially in the case of proper nouns (for instance, when the proper nouns starts with 'f' or 'k' which resemble clitics), or inconsistencies in segmentation between the treebank text and MADAMIRA analyses. For instance, the word لَالَّا "l{lA" (Eng.: so that not), in the MADAMIRA analysis is represented as لـ أن لَا "l >n lA", i.e., the word is segmented, causing a token mismatch. The same happens for the word لكي "lky" (Eng.: so) in MADAMIRA, which is represented as "l ky" in the treebank. These are idiosyncratic differences that need to be handled on a word by word basis. There are also few misspelled words in the source text that are not recognized by MADAMIRA.

While we used normalization, and found it to be the best option to reduce the number of sentences without errors, we believe the case of Hamza should be handled differently. This was also examined by Maamouri et al. (2008), who show that normalizing the dataset is not enough.

6 Conclusion

The work described in this paper describes a situation where we parse Arabic in a realistic setting, i.e, where we do not assume treebank segmentation, but instead use a state-of-the–art segmenter and morphological analyzer to obtain segmentation. Since we need a graph of all possible analyses for our future research, we need to use the vocalized forms without being able to use specific tokenization schemes that MADAMIRA offers. The numbers presented in section 4.4 show very clearly that we need to be extremely careful in how we handle differences in orthography and segmentation. These differences are partly due to segmentation ambiguities that cannot be resolved without access to syntactic information. These are the issues that we were originally interested in. However, such cases cases are overshadowed by a wide range of other cases, which need to be resolved in order to be able to focus on the interesting cases. The cases that need to be handled before we can even seriously start thinking about parsing experiments including different decisions in segmentation such as the case of the definite determiner and the Ta-Marbuta, but also orthographic inconsistencies involving Hamza and Alif. All of those decisions need to be explained in detail in order to ensure that the parsing research is replicable.

For the future, we plan to investigate a lattice approach to parsing. In this setting, the parser will be provided with a set of analyses from MADAMIRA, integrated into a lattice, which will allow the parser to choose the segmentation that works best given the syntactic constraints

of the grammar. This opens the question of how many analyses we should give the parser, since having access to too many irrelevant analyses may be more of a hindrance than helpful. This was shown in work on integrating word segmentation and parsing for Chinese (Hu et al., 2017). Other issues concerns the weighting of the arcs in the lattice, which have been shown to be useful in Arabic lattice parsing (Green and Manning, 2010), the usefulness of automatic morphological analyses, and the effect of normalization on parsing results.

Acknowledgments

We are grateful to Djamé Seddah for giving us his feedback on the experiments and the interpretation of the results. We are also grateful to the anonymous reviewers for their insightful comments.

References

Al-Emran, M., Zaza, S., and Shaalan, K. (2015). Parsing Modern Standard Arabic using treebank resources. In *2015 International Conference on Information and Communication Technology Research (ICTRC)*, pages 80–83.

Attia, M., Foster, J., Hogan, D., Roux, J. L., Tounsi, L., and Van Genabith, J. (2010). Handling unknown words in statistical latent-variable parsing models for Arabic, English and French. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 67–75. Association for Computational Linguistics.

Becker, M. and Frank, A. (2002). A stochastic topological parser for German. In *Proceedings of the 19th Onternational Conference on Computational Linguistics*, pages 1–7.

Bod, R. (1996). Monte Carlo Parsing. In Bunt, H. and Tomita, M., editors, *Recent Advances in Parsing Technology*, pages 255–280. Kluwer.

Bod, R. (2001). What is the minimal set of fragments that achieves maximal parse accuracy? In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 66–73.

Buckwalter, T. (2004). Arabic morphological analyzer version 2.0. Linguistic Data Consortium.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 173–180.

Cheung, J. C. K. and Penn, G. (2009). Topological field parsing of German. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 64–72.

Chiang, D., Diab, M., Habash, N., Rambow, O., and Shareef, S. (2006). Parsing Arabic dialects. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Collins, M. and Koo, T. (2005). Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

Dakota, D. and Kübler, S. (2017). Towards replicability in parsing. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 185–194, Varna, Bulgaria.

Diab, M. (2009). Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking. In *2nd International Conference on Arabic Language Resources and Tools*, volume 110.

Diab, M., Hacioglu, K., and Jurafsky, D. (2004). Automatic tagging of Arabic text: From raw text to base phrase chunks. In *Proceedings of HLT-NAACL 2004: Short papers*, pages 149–152.

Goldberg, Y. and Elhadad, M. (2011). Joint Hebrew segmentation and parsing using a PCFG-LA lattice parser. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 704–709.

Goodman, J. (1996). Efficient algorithms for parsing the DOP model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA.

Green, S. and Manning, C. D. (2010). Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 394–402.

Habash, N. (2010). *Introduction to Arabic Natural Language Processing*, volume 3 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers.

Habash, N. and Rambow, O. (2005). Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 573–580.

Habash, N., Rambow, O., and Roth, R. (2009). MADA+ TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, volume 41, Cairo, Egypt.

Habash, N. and Roth, R. M. (2009). Catib: The Columbia Arabic Treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference*, pages 221–224.

Habash, N. and Sadat, F. (2006). Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 49–52.

Hu, H., Dakota, D., and Kübler, S. (2017). Non-deterministic segmentation for Chinese lattice parsing. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, Varna, Bulgaria.

Johnson, M. (1998). PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430.

Kulick, S. and Bies, A. (2009). Treebank analysis and search using an extracted tree grammar. In *Eighth International Workshop on Treebanks and Linguistic Theories*.

Maamouri, M., Bies, A., Buckwalter, T., and Mekki, W. (2004). The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, volume 27, pages 466–467, Cairo, Egypt.

Maamouri, M., Kulick, S., and Bies, A. (2008). Diacritic annotation in the Arabic treebank and its impact on parser evaluation. In *LREC*.

McClosky, D., Charniak, E., and Johnson, M. (2006). Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 337–344.

Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *LREC*, volume 14, pages 1094–1101.

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440.

Petrov, S. and Klein, D. (2007). Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 404–411.

Petrov, S. and McDonald, R. (2012). Overview of the 2012 Shared Task on Parsing the Web. In *SANCL*, Montreal, Canada.

Seddah, D., Kübler, S., and Tsarfaty, R. (2014). Introducing the SPMRL 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages (SPMRL-SANCL)*, pages 103–109, Dublin, Ireland.

Seddah, D., Sagot, B., and Candito, M. (2012). The Alpage architecture at the SANCL 2012 shared task: robust pre-processing and lexical bridging for user-generated content parsing. In *SANCL 2012-First Workshop on Syntactic Analysis of Non-Canonical Language, an NAACL-HLT'12 workshop*.

Seddah, D., Tsarfaty, R., Kübler, S., Candito, M., Choi, J. D., Farkas, R., Foster, J., Goenaga, I., Gojenola Galletebeitia, K., Goldberg, Y., Green, S., Habash, N., Kuhlmann, M., Maier, W., Nivre, J., Przepiórkowski, A., Roth, R., Seeker, W., Versley, Y., Vincze, V., Woliński, M., Wróblewska, A., and de la Clergerie, E. V. (2013). Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, WA.

Sekine, S. and Collins, M. (1997). EVALB bracket scoring program. *URL: http://www. cs. nyu. edu/cs/projects/proteus/evalb*.

Wagner, J., Seddah, D., Foster, J., and Van Genabith, J. (2007). C-structures and F-structures for the British National Corpus. In *Proceedings of the Twelfth International Lexical Functional Grammar Conference*. CSLI Publications.