

Noun Phrase Chunking in Hebrew Influence of Lexical and Morphological Features

Yoav Goldberg and Meni Adler and Michael Elhadad

Computer Science Department

Ben Gurion University of the Negev

P.O.B 653 Be'er Sheva 84105, Israel

{yoavg, adlerm, elhadad}@cs.bgu.ac.il

Abstract

We present a method for Noun Phrase chunking in Hebrew. We show that the traditional definition of base-NPs as non-recursive noun phrases does not apply in Hebrew, and propose an alternative definition of Simple NPs. We review syntactic properties of Hebrew related to noun phrases, which indicate that the task of Hebrew SimpleNP chunking is harder than base-NP chunking in English. As a confirmation, we apply methods known to work well for English to Hebrew data. These methods give low results (F from 76 to 86) in Hebrew. We then discuss our method, which applies SVM induction over lexical and morphological features. Morphological features improve the average precision by $\sim 0.5\%$, recall by $\sim 1\%$, and F-measure by ~ 0.75 , resulting in a system with average performance of 93% precision, 93.4% recall and 93.2 F-measure.*

1 Introduction

Modern Hebrew is an agglutinative Semitic language, with rich morphology. Like most other non-European languages, it lacks NLP resources and tools, and specifically there are currently no available syntactic parsers for Hebrew. We address the task of NP chunking in Hebrew as a

first step to fulfill the need for such tools. We also illustrate how this task can successfully be approached with little resource requirements, and indicate how the method is applicable to other resource-scarce languages.

NP chunking is the task of labelling noun phrases in natural language text. The input to this task is free text with part-of-speech tags. The output is the same text with brackets around base noun phrases. A base noun phrase is an NP which does not contain another NP (it is not recursive). NP chunking is the basis for many other NLP tasks such as shallow parsing, argument structure identification, and information extraction

We first realize that the definition of base-NPs must be adapted to the case of Hebrew (and probably other Semitic languages as well) to correctly handle its syntactic nature. We propose such a definition, which we call *simple NPs* and assess the difficulty of chunking such NPs by applying methods that perform well in English to Hebrew data. While the syntactic problem in Hebrew is indeed more difficult than in English, morphological clues do provide additional hints, which we exploit using an SVM learning method. The resulting method reaches performance in Hebrew comparable to the best results published in English.

2 Previous Work

Text chunking (and NP chunking in particular), first proposed by Abney (1991), is a well studied problem for English. The CoNLL2000 shared task (Tjong Kim Sang et al., 2000) was general chunking. The best result achieved for the shared task data was by Zhang et al (2002), who achieved NP chunking results of 94.39% precision, 94.37% recall and 94.38 F-measure using a

* This work was funded by the Israel Ministry of Science and Technology under the auspices of the Knowledge Center for Processing Hebrew. Additional funding was provided by the Lynn and William Frankel Center for Computer Sciences.

generalized Winnow algorithm, and enhancing the feature set with the output of a dependency parser. Kudo and Matsumoto (2000) used an SVM based algorithm, and achieved NP chunking results of 93.72% precision, 94.02% recall and 93.87 F-measure for the same shared task data, using only the words and their PoS tags. Similar results were obtained using Conditional Random Fields on similar features (Sha and Pereira, 2003).

The NP chunks in the shared task data are base-NP chunks – which are non-recursive NPs, a definition first proposed by Ramshaw and Marcus (1995). This definition yields good NP chunks for English, but results in very short and uninformative chunks for Hebrew (and probably other Semitic languages).

Recently, Diab et al (2004) used SVM based approach for Arabic text chunking. Their chunks data was derived from the LDC Arabic TreeBank using the same program that extracted the chunks for the shared task. They used the same features as Kudo and Matsumoto (2000), and achieved over-all chunking performance of 92.06% precision, 92.09% recall and 92.08 F-measure (The results for NP chunks alone were not reported). Since Arabic syntax is quite similar to Hebrew, we expect that the issues reported below apply to Arabic results as well.

3 Hebrew Simple NP Chunks

The standard definition of English base-NPs is any noun phrase that does not contain another noun phrase, with possessives treated as a special case, viewing the possessive marker as the first word of a new base-NP (Ramshaw and Marcus, 1995). To evaluate the applicability of this definition to Hebrew, we tested this definition on the Hebrew TreeBank (Sima'an et al, 2001) published by the Hebrew Knowledge Center. We extracted all base-NPs from this TreeBank, which is similar in genre and contents to the English one. This results in extremely simple chunks.

| | English BaseNPs | Hebrew BaseNPs | Hebrew SimpleNPs |
|----------------|-----------------|----------------|------------------|
| Avg # of words | 2.17 | 1.39 | 2.49 |
| % length 1 | 30.95 | 63.32 | 32.83 |
| % length 2 | 39.35 | 35.48 | 32.12 |
| % length 3 | 18.68 | 0.83 | 14.78 |
| % length 4 | 6.65 | 0.16 | 9.47 |
| % length 5 | 2.70 | 0.16 | 4.56 |
| % length > 5 | 1.67 | 0.05 | 6.22 |

Table 1. Size of Hebrew and English NPs

Table 1 shows the average number of words in a base-NP for English and Hebrew. The Hebrew chunks are basically one-word groups around Nouns, which is not useful for any practical purpose, and so we propose a new definition for Hebrew NP chunks, which allows for some nestedness. We call our chunks Simple NP chunks.

3.1 Syntax of NPs in Hebrew

One of the reasons the traditional base-NP definition fails for the Hebrew TreeBank is related to syntactic features of Hebrew – specifically, *smixut* (construct state – used to express noun compounds), definite marker and the expression of possessives. These differences are reflected to some extent by the tagging guidelines used to annotate the Hebrew Treebank and they result in trees which are in general less flat than the Penn TreeBank ones.

Consider the example base noun phrase [*The homeless people*]. The Hebrew equivalent is

(1) האנשים מחוסרי הבית

which by the non-recursive NP definition will be bracketed as:

[האנשים מחוסרי הבית], or, loosely translating back to English: [*the home*]*less* [*people*].

In this case, the fact that the bound-morpheme *less* appears as a separate construct state word with its own definite marker (*ha-*) in Hebrew would lead the chunker to create two separate NPs for a simple expression. We present below syntactic properties of Hebrew which are relevant to NP chunking. We then present our definition of Simple NP Chunks.

Construct State: The Hebrew genitive case is achieved by placing two nouns next to each other. This is called “noun construct”, or *smixut*. The semantic interpretation of this construct is varied (Netzer and Elhadad, 1998), but it specifically covers possession. The second noun can be treated as an adjective modifying the next noun. The first noun is morphologically marked in a form known as the *construct form* (denoted by *const*). The definite article marker is placed on the second word of the construction:

(2) בית ספר

beit sefer / house-[const] book

School

(3) בית הספר

beit ha-sefer / house-[const] the-book

The school

The construct form can also be embedded:

(4) משרד ראש הממשלה

misrad ro\$ ha-mem\$ala
Office-[const poss] head-[const] the-government
The prime-minister's office

Possessive: the *smixut* form can be used to indicate possession. Other ways to express possession include the possessive marker של - '\$el' / 'of' - (5), or adding a possessive suffix on the noun (6). The various forms can be mixed together, as in (7):

(5) הבית שלי
ha-bait \$el-i / the-house of-[poss 1st person]
My house
(6) ביתי
beit-i / house-[poss 1st person]
My house
(7) משרדו של ראש הממשלה
misrad-o \$el ro\$ ha-mem\$ala
Office-[poss 3rd] of head-[const] the-government
The prime minister office

Adjective: Hebrew adjectives come after the noun, and agree with it in number, gender and definite marker:

(8) התפוח הירוק
ha-tapu'ah ha-yarok / the-Apple the-green
The green apple

Some aspects of the predicate structure in Hebrew directly affect the task of NP chunking, as they make the decision to “split” NPs more or less difficult than in English.

Word order and the preposition 'et': Hebrew sentences can be either in SVO or VSO form. In order to keep the object separate from the subject, definite direct objects are marked with the special preposition 'et', which has no analog in English.

Possible null equative: The equative form in Hebrew can be null. Sentence (9) is a non-null equative, (10) a null equative, while (11) and (12) are predicative NPs, which look very similar to the null-equative form:

(9) הבית הוא גדול
ha-bait hu gadol
The-house is big
The house is big

(10) הבית גדול
ha-bait gadol
The-house big
The house is big

(11) בית גדול
bait gadol
House big
A big house

(12) הבית הגדול
ha-bait ha-gadol
The-house the-big
The big house

Morphological Issues: In Hebrew morphology, several lexical units can be concatenated into a single textual unit. Most prepositions, the definite article marker and some conjunctions are concatenated as prefixes, and possessive pronouns and some adverbs are concatenated as suffixes. The Hebrew Treebank is annotated over a segmented version of the text, in which prefixes and suffixes appear as separate lexical units. On the other hand, many bound morphemes in English appear as separate lexical units in Hebrew. For example, the English morphemes *re-*, *ex-*, *un-*, *-less*, *-like*, *-able*, appear in Hebrew as separate lexical units – מחדש, לשעבר, בלתי, לא/אי/בלתי, לדמוי, חסר, בלתי, לא, בר, חסרי, בלתי, לא.

In our experiment, we use as input to the chunker the text after it has been morphologically disambiguated and segmented. Our analyzer provides segmentation and PoS tags with 92.5% accuracy and full morphology with 88.5% accuracy (Adler and Elhadad, 2006).

3.2 Defining Simple NPs

Our definition of Simple NPs is pragmatic. We want to tag phrases that are complete in their syntactic structure, avoid the requirement of tagging recursive structures that include full clauses (relative clauses for example) and in general, tag phrases that have a simple denotation. To establish our definition, we start with the most complex NPs, and break them into smaller parts by stating what should **not** appear inside a Simple NP. This can be summarized by the following table:

| Outside SimpleNP | Exceptions |
|---|---|
| Prepositional Phrases | % related PPs are allowed: 5% מ המכירות 5% of the sales |
| Relative Clauses | |
| Verb Phrases | |
| Apposition ¹ | Possessive של - '\$el' / 'of' - is not considered a PP |
| Some conjunctions (Conjunctions are marked according to the TreeBank guidelines) ² . | |

Table 2. Definition of Simple NP chunks

Examples for some Simple NP chunks resulting from that definition:

¹ Apposition structure is not annotated in the TreeBank. As a heuristic, we consider every comma inside a non conjunctive NP which is not followed by an adjective or an adjective phrase to be marking the beginning of an apposition.

² As a special case, Adjectival Phrases and possessive conjunctions are considered to be inside the Simple NP.

[תופעה זו] התבררה אתמול ב[וועדת העבודה והרווחה של הכנסת] שדנה ב[נושא העסקת עובדים זרים] [This phenomenon] was highlighted yesterday at [the labor and welfare committee-const of the Knesset] that dealt with [the topic-const of foreign workers employment-const].

[המעסיקים] אינם מצפים שיצליחו למשוך [מספר ניכר של עובדים ישראליים] ל[קטיף] בגלל [השכר הנמוך] המשולם³ ל[עבודה זו]. [The employers] do not expect to succeed in attracting [a significant number of Israeli workers] for [the fruit-picking] because of [the low salaries] paid for [this work].

This definition can also yield some rather long and complex chunks, such as:

[כיבושיהם של גינגיס חאן והצבא המונגולי הטטרי שליו] [The conquests of Genghis Khan and his Mongol Tartar army]

ל [דברי פקידי ממשלה מקומיים] , [מפעלים] על [שטח טטרי] הרוויחו ב [ה שנה] ש עברה [סך של 3 . 7 מיליארד רובל (2 . 2 מיליארד דולר)] , ש [את כולם כמעט] לקחה [מוסקבה]

According to [reports of local government officials], [factories] on [Tartar territory] earned in [the year] that passed [a sum of 3.7 billion Rb (2.2 billion dollars)], which [Moscow] took [almost all].

Note that Simple NPs are split, for example, by the preposition ‘on’ ([factories] on [Tartar territory]), and by a relative clause ([a sum of 3.7Bn Rb] which [Moscow] took [almost all]).

3.3 Hebrew Simple NPs are harder than English base NPs

The Simple NPs derived from our definition are highly coherent units, but are also more complex than the non-recursive English base NPs. As can be seen in Table 1, our definition of Simple NP yields chunks which are on average considerably longer than the English chunks, with about 20% of the chunks with 4 or more words (as opposed to about 10% in English) and a significant portion (6.22%) of chunks with 6 or more words (1.67% in English).

Moreover, the baseline used at the CoNLL shared task⁴ (selecting the chunk tag which was most frequently associated with the current PoS)

³ For readers familiar with Hebrew and feel that המשולם is an adjective and should be inside the NP, we note that this is not the case – המשולם here is actually a Verb in the Beinoni form and the definite marker is actually used as relative marker.

⁴ <http://www.cnts.ua.ac.be/conll2000/chunking/>

gives far inferior results for Hebrew SimpleNPs (see Table 3).

4 Chunking Methods

4.1 Baseline Approaches

We have experimented with different known methods for English NP chunking, which resulted in poor results for Hebrew. We describe here our experiment settings, and provide the best scores obtained for each method, in comparison to the reported scores for English.

All tests were done on the corpus derived from the Hebrew Tree Bank. The corpus contains 5,000 sentences, for a total of 120K tokens (agglutinated words) and 27K NP chunks (more details on the corpus appear below). The last 500 sentences were used as the test set, and all the other sentences were used for training. The results were evaluated using the CoNLL shared task evaluation tools⁵. The approaches tested were Error Driven Pruning (EDP) (Cardie and Pierce, 1998) and Transformational Based Learning of IOB tagging (TBL) (Ramshaw and Marcus, 1995).

The Error Driven Pruning method does not take into account lexical information and uses only the PoS tags. For the Transformation Based method, we have used both the PoS tag and the word itself, with the same templates as described in (Ramshaw and Marcus, 1995). We tried the Transformational Based method with more features than just the PoS and the word, but obtained lower performance. Our best results for these methods, as well as the CoNLL baseline (BASE), are presented in Table 3. These results confirm that the task of Simple NP chunking is harder in Hebrew than in English.

4.2 Support Vector Machines

We chose to adopt a tagging perspective for the Simple NP chunking task, in which each word is to be tagged as either B, I or O depending on whether it is in the Beginning, Inside, or Outside of the given chunk, an approach first taken by Ramshaw and Marcus (1995), and which has become the *de-facto* standard for this task. Using this tagging method, chunking becomes a classification problem – each token is predicted as being either I, O or B, given features from a predefined linguistic context (such as the

⁵ http://www.cnts.ua.ac.be/conll2000/chunking/conllev_al.txt

words surrounding the given word, and their PoS tags).

One model that allows for this prediction is Support Vector Machines - SVM (Vapnik, 1995). SVM is a supervised machine learning algorithm which can handle gracefully a large set of overlapping features. SVMs learn binary classifiers, but the method can be extended to multi-class classification (Allwein et al., 2000; Kudo and Matsumoto, 2000).

SVMs have been successfully applied to many NLP tasks since (Joachims, 1998), and specifically for base phrase chunking (Kudo and Matsumoto, 2000; 2003). It was also successfully used in Arabic (Diab et al., 2004).

The traditional setting of SVM for chunking uses for the context of the token to be classified a window of two tokens around the word, and the features are the PoS tags and lexical items (word forms) of all the tokens in the context. Some settings (Kudo and Matsumoto, 2000) also include the IOB tags of the two “previously tagged” tokens as features (see Fig. 1).

This setting (including the last 2 IOB tags) performs nicely for the case of Hebrew Simple NPs chunking as well.

Linguistic features are mapped to SVM feature vectors by translating each feature such as “PoS at location n-2 is NOUN” or “word at location n+1 is DOG” to a unique vector entry, and setting this entry to 1 if the feature occurs, and 0 otherwise. This results in extremely large yet extremely sparse feature vectors.

| Method | English BaseNPs | | Hebrew SimpleNPs | | |
|--------|-----------------|-------|------------------|------|-------|
| | Prec | Rec | Prec | Rec | F |
| BASE | 72.58 | 82.14 | 64.7 | 75.4 | 69.78 |
| EDP | 92.7 | 93.7 | 74.6 | 78.1 | 76.3 |
| TBL | 91.3 | 91.8 | 84.7 | 87.7 | 86.2 |

Table 3. Baseline results for Simple NP chunking SVM Chunking in Hebrew

| WORD | POS | CHUNK | |
|-------|------|-------|---------------|
| ה | NA | B-NP | |
| מאבק | NOUN | I-NP | Feature Set |
| בין | PREP | O | |
| רוסיה | NAME | B-NP | Estimated Tag |
| ל | PREP | O | |
| ה | NA | B-NP | |
| טורים | NOUN | I-NP | |

Figure 1. Linguistic features considered in the basic SVM setting for Hebrew

4.3 Augmentation of Morphological Features

Hebrew is a morphologically rich language. Recent PoS taggers and morphological analyzers for Hebrew (Adler and Elhadad, 2006) address this issue and provide for each word not only the PoS, but also full morphological features, such as Gender, Number, Person, Construct, Tense, and the affixes' properties. Our system, currently, computes these features with an accuracy of 88.5%.

Our original intuition is that the difficulty of Simple NP chunking can be overcome by relying on morphological features in a small context. These features would help the classifier decide on agreement, and split NPs more accurately.

Since SVMs can handle large feature sets, we utilize additional morphological features. In particular, we found the combination of the Number and the Construct features to be most effective in improving chunking results. Indeed, our experiments show that introducing morphological features improves chunking quality by as much as 3-point in F-measure when compared with lexical and PoS features only.

5 Experiment

5.1 The Corpus

The Hebrew TreeBank⁶ consists of 4,995 hand annotated sentences from the *Ha'aretz* newspaper. Besides the syntactic structure, every word is PoS annotated, and also includes morphological features. The words in the TreeBank are segmented: *ב ה בית של אנהו* (instead of *בבית שלנו*). Our morphological analyzer also provides such segmentation.

We derived the Simple NPs structure from the TreeBank using the definition given in Section 3.2. We then converted the original Hebrew TreeBank tagset to the tagset of our PoS tagger. For each token, we specify its word form, its PoS, its morphological features, and its correct IOB tag. The result is the Hebrew Simple NP chunks corpus⁷. The corpus consists of 4,995 sentences, 27,226 chunks and 120,396 segmented tokens. 67,919 of these tokens are covered by NP chunks. A sample annotated sentence is given in Fig. 2.

⁶<http://mila.cs.technion.ac.il/website/english/resources/corpora/treebank/index.html>

⁷<http://www.cs.bgu.ac.il/~nlpproj/chunking>

| | | | | | | | | | | | |
|-------|-------------|----|----|---|----|---|-------|---|----|----|------|
| ב | PREPOSITION | NA | NA | N | NA | N | NA | N | NA | NA | O |
| ה | DEF_ART | NA | NA | N | NA | N | NA | N | NA | NA | B-NP |
| עבר | NOUN | M | S | N | NA | N | NA | N | NA | NA | I-NP |
| היה | AUXVERB | M | S | N | 3 | N | PAST | N | NA | NA | O |
| קל | ADJECTIVE | M | S | N | NA | N | NA | N | NA | NA | O |
| יותר | ADVERB | NA | NA | N | NA | N | NA | N | NA | NA | O |
| לקרוא | VERB | NA | NA | N | NA | Y | TOINF | N | NA | NA | O |
| את | ET_PREP | NA | NA | N | NA | N | NA | N | NA | NA | B-NP |
| ה | DEF_ART | NA | NA | N | NA | N | NA | N | NA | NA | I-NP |
| מפה | NOUN | F | S | N | NA | N | NA | N | NA | NA | I-NP |
| . | PUNCTUATION | NA | NA | N | NA | N | NA | N | NA | NA | O |

Figure 2. A Sample annotated sentence

5.2 Morphological Features:

The PoS tagset we use consists of 22 tags:

| | | |
|--------------|-------------|--------------|
| ADJECTIVE | ADVERB | ET_PREP |
| AUXVERB | CONJUNCTION | DEF_ART |
| DETERMINER | EXISTENTIAL | INTERJECTION |
| INTEROGATIVE | MODAL | NEGATION |
| PARTICLE | NOUN | NUMBER |
| PRONOUN | PREFIX | PREPOSITION |
| UNKNOWN | PROPERNAME | PUNCTUATION |
| VERB | | |

For each token, we also supply the following morphological features (in that order):

| Feature | Possible Values |
|---------------|--|
| Gender | (M)ale, (F)emale, (B)oth (unmarked case), (NA) |
| Number | (S)ingle, (P)lurar, (D)ual, can be (ALL), (NA) |
| Construct | (Y)es, (N)o |
| Person | (1)st, (2)nd, (3)rd, (123)all, (NA) |
| To-Infinitive | (Y)es, (N)o |
| Tense | Past, Present, Future, Beinoni, Imperative, ToInf, BareInf |
| (has) Suffix | (Y)es, (N)o |
| Suffix-Num | (M)ale, (F)emale, (B)oth, (NA) |
| Suffix-Gen | (S)ingle, (P)lurar, (D)ual, (DP)-dual plural, can be (ALL), (NA) |

As noted in (Rambow and Habash 2005), one cannot use the same tagset for a Semitic language as for English. The tagset we have derived has been extensively validated through manual tagging by several testers and cross-checked for agreement.

5.3 Setup and Evaluation

For all the SVM chunking experiments, we use the YamCha⁸ toolkit (Kudo and Matsumoto, 2003). We use forward moving tagging, using standard SVM with polynomial kernel of degree 2, and C=1. For the multiclass classification, we

⁸ <http://chasen.org/~taku/software/yamcha/>

use pairwise voting. For all the reported experiments, we chose the context to be a $-2/+2$ tokens windows, centered at the current token.

We use the standard metrics of *accuracy* (% of correctly tagged tokens), *precision*, *recall* and *F-measure*, with the only exception of normalizing all punctuation tokens from the data prior to evaluation, as the TreeBank is highly inconsistent regarding the bracketing of punctuations, and we don't consider the exclusions/inclusions of punctuations from our chunks to be errors (*i.e.*, “[a book ,] [an apple]” “[a book] , [an apple]” and “[a book] [, an apple]” are all equivalent chunkings in our view).

All our development work was done with the first 500 sentences allocated for testing, and the rest for training. For evaluation, we used a 10-fold cross-validation scheme, each time with different consecutive 500 sentences serving for testing and the rest for training.

5.4 Features Used

We run several SVM experiments, each with the settings described in section 5.3, but with a different feature set. In all of the experiments the two previously tagged IOB tags were included in the feature set. In the first experiment (denoted WP) we considered the word and PoS tags of the context tokens to be part of the feature set.

In the other experiments, we used different subsets of the morphological features of the tokens to enhance the features set. We found that good results were achieved by using the *Number* and *Construct* features together with the word and PoS tags (we denote this WPNC). Bad results were achieved when using all the morphological features together. The usefulness of feature sets was stable across all tests in the ten-fold cross validation scheme.

5.5 Results

We discuss the results of the WP and WPNC experiments in details, and also provide the results for the WPG (using the *Gender* feature), and ALL (using *all* available morphological features) experiments, and P (using only PoS tags).

As can be seen in Table 4, lexical information is very important: augmenting the PoS tag with lexical information boosted the F-measure from 77.88 to 92.44. The addition of the extra morphological features of *Construct* and *Number* yields another increase in performance, resulting in a final F-measure of 93.2%. Note that the effect of these morphological features on the overall accuracy (the number of BIO tagged cor-

rectly) is minimal (Table 5), yet the effect on the precision and recall is much more significant. It is also interesting to note that the Gender feature hurts performance, even though Hebrew has agreement on both Number and Gender. We do not have a good explanation for this observation – but we are currently verifying the consistency of the gender annotation in the corpus (in particular, the effect of the unmarked gender tag).

We performed the WP and WPNC experiment on two forms of the corpus: (1) WP, WPNC using the manually tagged morphological features included in the TreeBank and (2) WPE, WPNC using the results of our automatic morphological analyzer, which includes about 10% errors (both in PoS and morphological features). With the manual morphology tags, the final F-measure is 93.20, while it is 91.40 with noise. Interestingly, the improvement brought by adding morphological features to chunking in the noisy case (WPNC) is almost 3.0 F-measure points (as opposed to 0.758 for the "clean" morphology case WPNC).

| Features | Acc | Prec | Rec | F |
|----------|--------------|--------------|--------------|--------------|
| P | 91.77 | 77.03 | 78.79 | 77.88 |
| WP | 97.49 | 92.54 | 92.35 | 92.44 |
| WPE | 94.87 | 89.14 | 87.69 | 88.41 |
| WPG | 97.41 | 92.41 | 92.22 | 92.32 |
| ALL | 96.68 | 90.21 | 90.60 | 90.40 |
| WPNC | 97.61 | 92.99 | 93.41 | 93.20 |
| WPNC | 96.99 | 91.49 | 91.32 | 91.40 |

Table 4. SVM results for Hebrew

| Features | Prec | Rec | F |
|----------|--------------|--------------|--------------|
| WPNC | 0.456 | 1.058 | 0.758 |
| WPNC | 2.35 | 3.60 | 2.99 |

Table 5. Improvement over WP

5.6 Error Analysis and the Effect of Morphological Features

We performed detailed error analysis on the WPNC results for the entire corpus. At the individual token level, Nouns and Conjunctions caused the most confusion, followed by Adverbs and Adjectives. Table 6 presents the confusion matrix for all POSs with a substantial amount of errors. I→O means that the correct chunk tag was I, but the system classified it as O. By examining the errors on the chunks level, we identified 7 common classes of errors:

Conjunction related errors: bracketing “[a] and [b]” instead of “[a and b]” and *vice versa*.

Split errors: bracketing [a][b] instead of [a b]

Merge errors: bracketing [a b] instead of [a][b]

Short errors: bracketing “a [b]” or “[a b]” instead of [a b]

Long errors: bracketing “[a b]” instead of “[a b]” or “[a [b]”

Whole Chunk errors: either missing a whole chunk, or bracketing something which doesn’t overlap with a chunk at all (extra chunk).

Missing/ExtraToken errors: this is a generalized form of conjunction errors: either “[a T [b]” instead of “[a T b]” or vice versa, where T is a single token. The most frequent of such words (other than the conjuncts) was 'של' - the possessive '\$el'.

| POS | All | O→I | O→B | I→O | I→B | B→O | B→I |
|-----------|-----|------------|-----------|------------|------------|-----------|------------|
| NOUN | 602 | 7 | 27 | 0 | 342 | 4 | 222 |
| CONJ | 405 | 146 | 5 | 232 | 1 | 21 | 0 |
| ADVERB | 306 | 87 | 32 | 104 | 10 | 68 | 5 |
| DEF_ART | 247 | 18 | 10 | 9 | 104 | 5 | 101 |
| ADJECTIVE | 215 | 140 | 0 | 58 | 0 | 16 | 1 |
| PROPNAME | 168 | 4 | 8 | 0 | 82 | 0 | 74 |
| NUMBER | 158 | 14 | 6 | 4 | 64 | 8 | 62 |
| PREP | 152 | 81 | 0 | 62 | 0 | 9 | 0 |
| PRONOUN | 99 | 2 | 27 | 3 | 24 | 12 | 31 |

Table 6. WPNC Confusion Matrix

The data in Table 6 suggests that Adverbs and Adjectives related errors are mostly of the “short” or “long” types, while the Noun (including proper names and pronouns) related errors are of the “split” or “merge” types.

The most frequent error type was conjunction related, closely followed by split and merge. Much less significant errors were cases of extra Adverbs or Adjectives at the end of the chunk, and missing adverbs before or after the chunk.

Conjunctions are a major source of errors for English chunking as well (Ramshaw and Marcus, 1995, Cardie and Pierce, 1998)⁹, and we plan to address them in future work. The split and merge errors are related to argument structure, which can be more complicated in Hebrew than in English, because of possible null equatives. The too-long and too-short errors were mostly attachment related. Most of the errors are related to linguistic phenomena that cannot be inferred by the localized context used in our SVM encoding. We examine the types of errors that the addition of

⁹ Although base-NPs are by definition non-recursive, they may still contain CCs when the coordinators are ‘trapped’: “[securities and exchange commission]” or conjunctions of adjectives.

Number and Construct features fixed. Table 7 summarizes this information.

| ERROR | WP | WPNC | # Fixed | % Fixed |
|--------------------|-----|------|---------|---------|
| CONJUNCTION | 256 | 251 | 5 | 1.95 |
| SPLIT | 198 | 225 | -27 | -13.64 |
| MERGE | 366 | 222 | 144 | 39.34 |
| LONG (ADJ AFTER) | 120 | 117 | 3 | 2.50 |
| EXTRA CHUNK | 89 | 88 | 1 | 1.12 |
| LONG (ADV AFTER) | 77 | 81 | -4 | -5.19 |
| SHORT (ADV AFTER) | 67 | 65 | 2 | 2.99 |
| MISSING CHUNK | 50 | 54 | -4 | -8.00 |
| SHORT (ADV BEFORE) | 53 | 48 | 5 | 9.43 |
| EXTRA ןש TOK | 47 | 47 | 0 | 0.00 |

Table 7. Effect of Number and Construct information on most frequent error classes

The error classes most affected by the number and construct information were split and merge – WPNC has a tendency of splitting chunks, which resulted in some unjustified splits, but compensates this by fixing over a third of the merging mistakes. This result makes sense – construct and local agreement information can aid in the identification of predicate boundaries. This confirms our original intuition that morphological features do help in identifying boundaries of NP chunks.

6 Conclusion and Future work

We have noted that due to syntactic features such as *smixut*, the traditional definition of base NP chunks does not translate well to Hebrew and probably to other Semitic languages. We defined the notion of Simple NP chunks instead. We have presented a method for identifying Hebrew Simple NPs by supervised learning using SVM, providing another evidence for the suitability of SVM to chunk identification.

We have also shown that using morphological features enhances chunking accuracy. However, the set of morphological features used should be chosen with care, as some features actually hurt performance.

Like in the case of English, a large part of the errors were caused by conjunctions – this problem clearly requires more than local knowledge. We plan to address this issue in future work.

References

Meni Adler and Michael Elhadad, 2006. Unsupervised Morpheme-based HMM for Hebrew Morphological Disambiguation. In *Proc. of COLING/ACL 2006*, Sidney.

Steven P. Abney. 1991. Parsing by Chunks. In Robert C. Berwick, Steven P. Abney, and Carol Tenny

editors, *Principle Based Parsing*. Kluwer Academic Publishers.

Erin L. Allwein, Robert E. Schapire, and Yoram Singer. 2000. Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. *Journal of Machine Learning Research*, 1:113-141.

Claire Cardie and David Pierce. 1998. Error-Driven Pruning of Treebank Grammars for Base Noun Phrase Identification. In *Proc. of COLING-98*, Montréal.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks, In *Proc. of HLT/NAACL 2004*, Boston.

Nizar Habash and Owen Rambow, 2005. Arabic Tokenization, Part-of-speech Tagging and Morphological Disambiguation in One Fell Swoop. In *Proc. of ACL 2005*, Ann Arbor.

Thorsten Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proc. of ECML-98*, Chemnitz.

Taku Kudo and Yuji Matsumoto. 2000. Use of Support Vector Learning for Chunk Identification. In *Proc. of CoNLL-2000 and LLL-2000*, Lisbon.

Taku Kudo and Yuji Matsumoto. 2003. Fast Methods for Kernel-Based Text Analysis. In *Proc. of ACL 2003*, Sapporo.

Yael Netzer-Dahan and Michael Elhadad, 1998. Generation of Noun Compounds in Hebrew: Can Syntactic Knowledge be Fully Encapsulated? In *Proc. of INLG-98*, Ontario.

Lance A. Ramshaw and Mitchel P. Marcus. 1995. Text Chunking Using Transformation-based Learning. In *Proc. of the 3rd ACL Workshop on Very Large Corpora*. Cambridge.

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman and N. Nativ, 2001. Building a Tree-bank of Modern Hebrew Text, in *Traitement Automatique des Langues* 42(2).

Fei Sha and Fernando Pereira. 2003. Shallow Parsing with Conditional Random Fields. *Technical Report CIS TR MS-CIS-02-35*, University of Pennsylvania.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proc. of CoNLL-2000 and LLL-2000*, Lisbon.

Vladimir Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, NY.

Tong Zhang, Fred Damerau and David Johnson. 2002. Text Chunking based on a Generalization of Winnow. *Journal of Machine Learning Research*, 2: 615-637.