Improved Arabic Base Phrase Chunking with a new enriched POS tag set

Mona T. Diab
Center for Computational Learning Systems
Columbia University
mdiab@cs.columbia.edu

Abstract

Base Phrase Chunking (BPC) or shallow syntactic parsing is proving to be a task of interest to many natural language processing applications. In this paper, A BPC system is introduced that improves over state of the art performance in BPC using a new part of speech tag (POS) set. The new POS tag set, ERTS, reflects some of the morphological features specific to Modern Standard Arabic. ERTS explicitly encodes definiteness, number and gender information increasing the number of tags from 25 in the standard LDC reduced tag set to 75 tags. For the BPC task, we introduce a more language specific set of definitions for the base phrase annotations. We employ a support vector machine approach for both the POS tagging and the BPC processes. The POS tagging performance using this enriched tag set, ERTS, is at 96.13% accuracy. In the BPC experiments, we vary the feature set along two factors: the POS tag set and a set of explicitly encoded morphological features. Using the ERTS POS tagset, BPC achieves the highest overall $F_{\beta=1}$ of 96.33% on 10 different chunk types outperforming the use of the standard POS tag set even when explicit morphological features are present.

1 Introduction

Base Phrase Chunking (BPC), also known as shallow syntactic parsing, is the process by which adjacent words are grouped together to form non-recursive chunks in a sentence. The base chunks form phrases such as verb phrases, noun phrases and adjective phrases, etc. An English example of base phrases is *[I]$_{NP}$ [would eat]$_{VP}$ [red luscious apples]$_{NP}$ [on Sundays]$_{PP}$*. The BPC task is proving to be an enabling step that is useful to many natural language processing (NLP) applications such as information extraction and semantic role labeling (Hacioglu & Ward, 2003). In English, these applications have shown robust relative performance when exploiting BPC when compared to using full syntactic parses. In general, BPC is appealing as an enabling technology since state of the art performance for BPC is higher ($F_{\beta=1}$ 95.48%) than that for full syntactic parsing ($F_{\beta=1}$ 90.02%) in English (Collins, 2000; Kudo & Matsumato, 2000). Moreover, since BPC had been cast as a classification problem by Ramshaw and Marcus (1995), the task is performed with greater efficiency and is easily portable to new languages in a supervised manner (Diab et al., 2004; Diab et al., 2007). For Arabic, BPC is especially interesting as it constitutes a viable alternative to full syntactic parsing due to the low performance in Arabic full syntactic parsing (labeled $F_{\beta=1}$ = ∼80% compared to $F_{\beta=1}$=91.44% for BPC) (Bikel, 2004; Diab et al., 2007; Kulick et al., 2006).

In this paper, we present a support vector machine (SVM) based supervised method for Arabic BPC. The new BPC system achieves an $F_{\beta=1}$ of 96.33% across 10 base phrase chunk types. We vary the feature sets along two different factors: the usage of explicit morphological features, and the use of different part of speech (POS) tag sets. We introduce a new enriched POS set for Arabic which comprises 75 POS tags. The new POS tag set enriches the standard reduced POS tag set (RTS), distributed

Proceedings of the 5th Workshop on Important Unresolved Matters, pages 89–96,
Prague, Czech Republic, June 2007.

with the LDC Arabic Treebank (ATB) (Maamouri et al., 2004), by adding definiteness, gender and number information to the basic RTS. We devise an automatic POS tagger based on the enriched tag set, ERTS. We use the same unified discriminative model approach to both POS tagging and BPC. The POS tagging results using ERTS are comparable to state of the art POS tagging using RTS with an accuracy of 96.13% for ERTS and 96.15% for RTS. However, using the ERTS as a feature in the BPC task, we note an overall significant increase of 1% absolute $F_{\beta=1}$ improvement over the usage of RTS alone. Moreover, we experiment with different explicit morphological features together with the different POS tag sets with no significant improvement.

The paper is laid out as follows: Section 2 discusses state of the art related work in the field of BPC in both English and Arabic; Section 3 illustrates our approach for both POS and BPC in detail; Section 4 presents the experimental setup, results and discussions.

2 Related Work

English BPC has made a lot of head way in recent years. It was first cast as a classification problem by Ramshaw and Marcus (1995), as a problem of NP chunking. Then it was extended to include other types of base phrases by Sang and Buchholz (2000) in the CoNLL 2000 shared task. Most successful approaches are based on machine learning techniques and sequence modeling of the different labels associated with the chunks. Both generative algorithms such as HMMs and multilevel Markov models, as well as, discriminative methods such as Support Vector Machines (SVM) and conditional random fields have been used for the BPC task. The closest relevant approach to the current investigation is the work of Kudo and Matsumato (2000) (KM00) on using SVMs and a sequence model for chunking. A la Ramshaw and Marcus (1995), they represent the words as a sequence of labeled words with *IOB* annotations, where the *B* marks a word at the beginning of a chunk, *I* marks a word inside a chunk, and *O* marks those words (and punctuation) that are outside chunks. The *IOB* annotation scheme for the English example described earlier is illustrated in Table 1.

word	IOB label
I	*B-NP*
would	*B-VP*
eat	*I-VP*
red	*B-NP*
luscious	*I-NP*
apples	*I-NP*
on	*B-PP*
Sundays	*I-PP*
.	*O*

Table 1: *IOB* annotation example

KM00 develop a sequence model, YAMCHA, over the labeled sequences of words.[1] YAMCHA is based on the TinySVM algorithm (Joachims, 1998). They use a degree 2 polynomial kernel. In their work on the task of English BPC, YAMCHA achieves an overall $F_{\beta=1}$ of 93.48% on five chunk types. They report improved results of 93.91% using a combined voting scheme (Kudo & Matsumato, 2000).

As far as Arabic BPC, Diab et al. (2004 & 2007) adopt the KM00 model for Arabic using YAMCHA. They cast the Arabic data from the ATB in the *IOB* annotation scheme. They use the reduced standard LDC tag set, RTS, as their only feature. Their system achieves an overall $F_{\beta=1}$ of 91.44% on 9 chunk types.

3 Current Approach

This paper is a significant extension to the Diab et al. (2007) work. Similar to other researchers in the area of BPC, we adopt a discriminative approach. A la Ramshaw and Marcus (1995), and Kudo and Matsumato (2000), we use the IOB tagging style for modeling and classification.

Various machine learning approaches have been applied to POS and BPC tagging, by casting them as classification tasks. Given a set of features extracted from the linguistic context, a classifier predicts the POS or BPC class of a token. SVMs (Vapnik, 1995) are one such supervised machine learning algorithm, with the advantages of: discriminative training, robustness and a capability to handle a large number of (overlapping) features with good generalization per-

[1] http://www.chasen.org/ taku/software/yamcha/

formance. Consequently, SVMs have been applied in many NLP tasks with great success (Joachims, 1998; Hacioglu & Ward, 2003).

We adopt a unified tagging perspective for both the POS tagging and the BPC tasks. We address them using the same SVM experimental setup which comprises a standard SVM as a multi-class classifier (Allwein et al., 2000).

A la KM00, we use the `YAMCHA` sequence model on the SVMs to take advantage of the context of the items being compared in a vertical manner in addition to the encoded features in the horizontal input of the vectors. Accordingly, in our different tasks, we define the notion of context to be a window of fixed size around the segment in focus for learning and tagging.

3.1 POS Tagging

Modern Standard Arabic is a rich morphological language, where words are explicitly marked for case, gender, number, definiteness, mood, person, voice, tense and other features. These morphological features are explicitly encoded in the full tag set provided in the `ATB`. These full morphological tags amount to over 2000 tag types (FULL). As expected, such morphological tags are not very useful for automatic statistical syntactic parsing since they are extremely sparse.[2] Hence, the LDC introduced the reduced tag set (RTS) of 25 tags. RTS masks case, mood, gender, person, definiteness for all categories. It maintains voice and tense for verbs, and some number information for nouns, namely, marking plural vs. singular for nouns and proper nouns. Therefore, in the process it masks duality for nouns and number for all adjectives. It should be noted, however, that it is extremely useful to have these morphological tags in order to induce features. There exists a system that produces the full morphological POS tag set, MADA, with very high accuracy, 96% (Habash & Rambow, 2005).

In this work, we introduce a new tag set that explicitly marks gender, number, and definiteness for nominals (*namely*, nouns, proper nouns, adjectives and pronouns). Verbs, particles, as well as, the person feature on pronouns, are not affected by this enrichment process, since neither person nor mood are explicitly encoded. Morphological case is also not explicitly encoded in ERTS. The new tag set, ERTS, is derived from the FULL tag set where there is an explicit specification in the tag itself for the different features to be encoded. We restricted ERTS to this set of features as they tend to be explicitly marked in the surface form of unvowelized Arabic text. In ERTS, definiteness is encoded with a present (D) or an absent one. Gender is encoded with an F or an M, corresponding to Fem and Masc, respectively. Number is encoded with (Du) for dual or an (S) for plurals or the absence of any marking for singular. For example, Table 2 illustrates some words with the FULL morphological tag and their corresponding RTS and ERTS definitions.[3]

Our approach: ERTS comprises 75 tags. For the current system, only 57 tags are instantiated. We develop a POS tagger based on this new set. We adopt the `YAMCHA` sequence model based on the `TinySVM` classifier. The tagger trained for ERTS tag set uses lexical features of +/-4 character n-grams from the beginning and end of a word in focus. The context for `YAMCHA` is defined as +/-2 words around the focus word. The words before the focus word are considered with their ERTS tags. The kernel is a polynomial degree 2 kernel. We adopt the one-vs-all approach for classification, where the tagged examples for one class are considered positive training examples and instances for other classes are considered negative examples. We present results and a brief discussion of the POS tagging performance in Section 4.

3.2 Base Phrase Chunking

In this task, we use a setup similar to that of Kudo & Matsumato (2000) and Diab et al. (2004 & 2007), with the *IOB* annotation representation: Inside *I* a phrase, Outside *O* a phrase, and Beginning *B* of a phrase. However, we designate 10 types of chunked phrases. The chunk phrases identified for Arabic are *ADJP, ADVP, CONJP, INTJP, NP, PP, PREDP, PRTP, SBARP, VP*. Thus the task is a one of 21 classification task (since there are *I* and *B* tags for each chunk phrase type, and a single *O* tag). The 21 *IOB* tags are listed: {*O, I-ADJP, B-ADJP, I-ADVP,*

[2] Dan Bikel, personal communication.

[3] All the romanized Arabic is presented in the Buckwalter transliteration scheme (Buckwalter, 2002).

		Gloss	*FULL*	*RTS*	*ERTS*
حصيلة	*HSylp*	'outcome'	NOUN+NSUFF_FEM_SG+CASE_IND_NOM	NN	NNF
نهائية	*nhA}yp*	'final'	ADJ+NSUFF_FEM_SG+CASE_IND_NOM	JJ	JJF
حادث	*HAdv*	'accident'	NOUN+CASE_DEF_ACC	NN	NNM
النار	*AlnAr*	'the-fire'	DET+NOUN+CASE_DEF_GEN	NN	DNNM
الجماعي	*AljmAEy*	'group'	DET+ADJ+CASE_DEF_GEN	JJ	DJJM
شخصين	*$xSyn*	'two persons'	NOUN+NSUFF_MASC_DU_GEN	NN	NNMDu

Table 2: Examples of POS tag sets RTS, ERTS and FULL

B-ADVP, I-CONJP, B-CONJP, I-INTJP, B-INTJP, I-NP, B-NP, I-PP, B-PP, I-PREDP, B-PREDP, I-PRTP, B-PRTP, I-SBARP, B-SBARP, I-VP, B-VP}.

The training data is derived from the ATB using the ChunkLink software.[4] ChunkLink flattens the tree to a sequence of base (non-recursive) phrase chunks with their *IOB* labels. For example, a token occurring at the beginning of a noun phrase is labeled as *B-NP*. The following Table 3 Arabic example illustrates the *IOB* annotation scheme:

Tags	*B-VP*	*B-NP*	*I-NP*	*O*
Arabic	وقع	مساء	الجمعة	.
Translit	wqE	msA'	AljmEp	.
Gloss	happened	night	the-Friday	.

Table 3: An Arabic *IOB* annotation example

Chunklink, however, is tailored for English syntactic structures. Hence, in order to train on reasonable Arabic chunks, we modify Chunklink's output using linguistic knowledge of Arabic syntactic structures. Some of the rules used are described below (we illustrate using ERTS for ease of exposition).

• **IDAFA**: This syntactic structure marks possession in Arabic. It is syntactically the case where an indefinite noun is followed by a definite one. The Chunklink output is modified to ensure that they form a single NP. Example: مساء الجمعة *msA' AljmEp* 'night of Friday' is *IOB* annotated as [*msA'* DNN B-NP, *AljmEp* DNNM I-NP].

• **NOUN-ADJ**: Nouns followed by adjectives and they agree in their morphological features of gender, number and definiteness form a single NP chunk. Example: حصيلة نهائية رسمية *HSylp nhA}yp rsmyp* 'final official outcome' is *IOB* annotated as [*HSylp* NNF B-NP, *nhA}yp* JJF I-NP, *rsmyp* JJF I-NP]

• **Pronouns**: This is an artifact of the ATB style of clitic tokenization. All pronouns, except in nominative position in the sentence such as *hw* and *hy*, are chunk internal.

• **Interjections**: If an interjection is followed by a noun, the noun is marked as internal to the interjective phrase.

• **Prepositional Phrases**: Nouns following prepositions are considered internal to the prepositional phrase and are *IOB* annotated, I-PP.

Phrase Types: The different phrase types are described as follows.

• **ADJP**: This is an adjectival phrase. The adjectival phrase could comprise a single adjective if mentioned in isolation such as جيدا *jydA* 'well', or multiple words such as قريبا جدا *qrybA jdA* 'very soon'. The latter is *IOB* annotated [*qrybA* B-ADJP] and [*jdA* I-ADJP], respectively.

• **ADVP**: This is an adverbial phrase. It may comprise a single adverb following a verb, such as سريعا *sryEA* 'quickly', or multiple words such as لكن ها *lkn hA* 'but she'.[5] The latter is *IOB* annotated [*lkn* B-ADVP] and [*hA* I-ADVP], respectively.

• **CONJP**: This chunk marks conjunctive phrases. We see single word CONJP when the conjunction appears before a verb phrase or a prepositional phrase such the conjunction و *w* 'and'. But we also have multiword conjunctive phrases when the conjunction is followed by a noun. For instance, و الفلسطينيون *w AlflsTynywn* 'and the Palestinians'

[4]http://ilk.uvt.nl/ sabine/chunklink

[5]This is a result of the ATB clitic tokenization style.

is *IOB* annotated [*w* B-CONJP] and [*AlflsTynywn* I-CONJP].

• **INTJP**: This is an interjective phrase. The interjective phrase could comprise a single interjection if mentioned in isolation such as نعم *nEm* 'yes'. Or with multiple words such as يا اخت *yA Axt* 'Oh sister', where it is *IOB* annotated [*yA* B-INTJP] and [*Axt* I-INTJP].

• **NP**: This is a noun phrase. It may comprise a single noun or multiple nouns or a noun and one or more adjectives. In this phrase type, we see typical noun adjective constructions as in الزفاف الجماعي *AlzfAf AljmAEy* 'the group wedding', where the initial noun is marked as [*AlzfAf* B-NP], and the folowing adjective is *IOB* annotated [*AljmAEy* I-NP]. We also encounter *idafa* constructions. For example, ملك الاردن *mlk AlArdn* 'king of Jordan', where *mlk* 'king' is an indefinite noun, and *AlArdn* 'Jordan' is a definite one. This phrase is *IOB* annotated [*mlk* B-NP] and [*AlArdn* I-NP].

• **PP**: This is a prepositional phrase. The phrase starts with a preposition followed by a pronoun, noun or proper noun. If the noun or proper noun is itself the beginning of an NP, the whole NP is internal to the PP. For example, خلال حفل الزفاف الجماعي *xlAl Hfl AlzfAf AljmAEy* 'during the group wedding party', *xlAl* is the preposition and is *IOB* annotated [*xlAl* B-PP], [*Hfl* I-PP] (if *Hfl* were not preceded by a preposition it would have been annotated [*Hfl* B-NP]), then for the noun [*AlzfAf* I-PP], and finally the adjective [*AljmAEy* I-PP].

• **PREDP**: This is a predicative phrase. It typically begins with a particle إن *An* '[is]', followed by a noun phrase. For example, إن الاصلاح الديني مهمة المجددين *An AlASlAH Aldyny mhmp Almjddyn* 'religious improvement is the reformers' task'. In our data, since we do not mark recursive structures in this BPC level, only the predicative particle is *IOB* annotated with B-PREDP. If it were followed by a possessive pronoun, the pronoun is annotated I-PREDP.

• **PRTP**: This phrase type marks particles such as negative particles that precede both nouns and verbs. A particle could be single word or a complex particle phrase. An example of a simple word particle is لم *lm* 'not', and a complex one is لا سيّما *lA sy∼mA* 'not as long'. In the latter case, it is *IOB* annotated [*lA* B-PRTP] and [*sy∼mA* I-PRTP].

• **SBARP**: This phrase structure marks the subjunctive constructions. SBARP phrases typically begin with a particle meaning 'that' such as أن *An* or مما *mmA* or الذي *Al*y* followed by a verb phrase.

• **VP**: This is a verb phrase. VP phrases are typically headed by a verb. All object pronouns proceeding a verb are *IOB* annotated I-VP. Moreover, we observe cases where a VP is headed by nominals (nouns and adjectives, in particular). The majority of these nominals are the active participle. The active participle in the `ATB` is tagged as an adjective. Active participles in Arabic are equivalent to predicative nominals in English (gerunds). Hence, some VPs are headed by JJs. An example active participle heading a verb phrase in our data is متهما *mthmA* 'accusing'.

Our Approach: We vary two factors in our feature sets: the POS tag set, and the presence or absence of explicit morphological features. We have three possible tag sets: RTS, ERTS and the full morphological tag set (FULL). We define a set of 6 morphological features (and their possible values): CASE (*ACC, GEN, NOM, NULL*), MOOD (*Indicative, Jussive, Subjunctive, NULL*), DEF (*DEF, INDEF, NULL*), NUM (*Sing, Dual, Plural, NULL*), GEN (*Fem, Masc, NULL*), PER (*1, 2, 3, NULL*).

From the intersection of the two factors, we devise 10 different experimental conditions. The conditions always have one of the POS tag sets and either no explicit features (noFeat), all explicit features (allFeat), or some selective features of: case mood and person (CASE_MOOD_PER), or definiteness gender and number (DEF_GEN_NUM). Therefore, the experimental conditions are as follows: RTS-noFeat, RTS-allFeat, RTS-CASE_MOOD_PER, RTS-DEF_GEN_NUM, ERTS-noFeat, ERTS-allFeat, ERTS-CASE_MOOD_PER, ERTS-DEF_GEN_NUM, FULL-noFeat, FULL-allFeat.

The BPC context is defined as a window of $+/-2$ tokens centered around the focus word where all the features for the specific condition are used and the tags for the previous two tokens before the focus token are also considered.

4 Experiments and Results

4.1 Data

The dev, test and training data are obtained from `ATB1v3`, `ATB2v2` and `ATB3v2` (Maamouri et al., 2004). We adopt the same data splits introduced by Chiang et. al (2006). The corpora are all news genre. The total development data comprises 2304 sentences and 70188 tokens, the total training data comprises 18970 sentences and 594683 tokens, and the total test data comprises 2337 sentences and 69665 tokens.

We use the unvocalized Buckwalter transliterated version of the `ATB`. For both POS tagging and BPC, we use the gold annotations of the training and test data for preprocessing. Hence, for POS tagging, the training and test data are both gold tokenized in the `ATB` clitic tokenization style. And for BPC, the POS tags, the morphological features, and, the tokenization is all gold. We derive the gold ERTS deterministically from the FULL set for the BPC results reported here.

The IOB annotations on the training and gold evaluation data are derived using `Chunklink` followed by our linguistic fixes described in Section 3.

4.2 SVM Setup

We use the default values for `YAMCHA` with the `C` parameter set to 0.5. It has a degree 2 polynomial kernel. `YAMCHA` adopts a one-vs-all binarization method.

4.3 Evaluation Metric

Standard metrics of Accuracy (Acc.), Precision, Recall, and F-measure $F_{\beta=1}$, on the test data are utilized. For both POS tagging and BPC, we use the `CoNLL` shared task evaluation tools.[6]

4.4 Results

4.4.1 ERTS POS Tagging Results

Table 4 shows the results obtained with the `YAMCHA` based POS tagger, *POS-TAG*, and the results obtained with a simple baseline, *BASELINE*. *BASELINE* is a supervised baseline, where the most frequent POS tag associated with a token from the training data is assigned to it in the test set, regardless of context. If the token does not occur in the training data, the token is assigned the *NN* tag as a default tag.

POS tagset	Acc.%
`RTS`	96.15
`ERTS`	96.13
`BASELINE`	86.5

Table 4: Results of `POS-TAG` on two different tag sets RTS and ERTS

`POS-TAG` clearly outperforms the most frequent baseline. Looking closely at the data, the worst obtained results are for the NO_FUNC category, as it is randomly confusable with almost all POS tags. Then, the imperative verbs are mostly confused with passive verbs 50% of the time, however the test data only comprises 8 imperative verbs. VBN, passive verbs, yields an accuracy of 68% only. It is worth noting that the most frequent baseline for VBN is 21%. VBN is a most difficult category to discern in the absence of the passivization diacritic which is naturally absent in unvowelized text (our experimental setup). The overall performance on the nouns and adjectives is relatively high. However, confusing these two categories is almost always present due to the inherent ambiguity. In fact, almost all Arabic adjectives could be used as nouns in Arabic.[7]

4.4.2 Base Phrase Chunking (BPC)

Table 5 illustrates the overall obtained results by our BPC system over the different experimental conditions.

The overall results for all the conditions significantly outperform state of the art published results on Arabic BPC of $F_{\beta=1}$=91.44% in Diab et al. (2004 & 2007). This is mainly attributed to the better quality annotations associated with tailoring of the `Chunklink` *IOB* annotations to the Arabic language characteristics.

All the $F_{\beta=1}$ results yielded by ERTS POS tag set outperform their counterparts using the RTS POS tagset. In fact, ERTS-noFeat condition outperforms all other conditions in our experiments.

We note that adding morphological features to the RTS POS tag set helps the performance slightly

[6] http://cnts.uia.ac.be/conll2003/ner/bin/conlleval

[7] This inherent ambiguity leads to inconsistency in the `ATB` gold annotations.

Condition	$F_{\beta=1}$	Condition	$F_{\beta=1}$
RTS-noFeat	95.41	ERTS-noFeat	**96.33**
RTS-CASE_MOOD_PER	95.73	ERTS-CASE_MOOD_PER	**96.32**
RTS-DEF_GEN_NUM	95.8	ERTS-DEF_GEN_NUM	**96.33**
RTS-allFeat	95.97	ERTS-allFeat	**96.25**
FULL-noFeat	96.29	FULL-allFeat	96.22

Table 5: Overall $F_{\beta=1}$ % results yielded for the different BPC experimental conditions

as we see a sequence of small jumps in performance from RTS-noFeat (95.41%) to RTS-allFeat (95.97%). However adding these features to the ERTS and FULL conditions does not help. In fact, in both the allFeat conditionsfor both ERTS and FULL, we note a slight decrease. The ERTS condition performance goes down from 96.33% (ERTS-noFeat) to 96.25% (ERTS-allFeat), and the FULL condition performance goes down from 96.29% (FULL-noFeat) to 96.22% (FULL-allFeat). This suggests that the features are not adding much information over and above what is already encoded in the POS tag set, and, in fact adding the explicit morphological features might be adding noise.

There is no significant difference between using ERTS and FULL in the overall results. However, we note that ERTS conditions slightly outperform the FULL conditions. This may be attributed the consistency introduced by ERTS over FULL, i.e., if FULL is not consistent in assigning CASE or MOOD or PER, for instance, ERTS, being insensitive to these features masks these inconsistencies present in the FULL tag set.

RTS-DEF_GEN_NUM may be viewed as an explicit encoding of the features in ERTS-noFeat, however, ERTS-noFeat outperforms it. Explicitly encoding the CASE MOOD and PER features does not help ERTS, in fact we see a slight drop in overall performance. However, upon closer inspection of the results per phrase type, we note slight relative improvement on PRTP and VP chunk types performance when the CASE, MOOD and PER are explicitly encoded. In ERTS-CASE_MOOD_PER, VP yields an $F_{\beta=1}$ of 99.3% and PRTP yields 97.2%, corresponding to ERTS-noFeat where VP yields an $F_{\beta=1}$ of 99.2% and PRTP an $F_{\beta=1}$ of 96.8%.

To better assess the quality of the performance and impact of the new POS tag set, we examine closely in Table 6 the phrase types directly affected by the added information whether encoded in the POS tag set or explicitly used as independent features. These phrase types are the ADJP, INTJP, NP and PP. The PP scored highly across the board with $F_{\beta=1}$ over 99% for all conditions, hence, it is not included in Table 6.

Condition	ADJP	INTJP	NP
RTS-noFeat	68.42	55.17	92.98
RTS-CASE_MOOD_PER	69.47	59.26	93.72
RTS-DEF_GEN_NUM	71.57	**64.29**	93.71
RTS-allFeat	72.22	57.14	94.15
ERTS-noFeat	72.35	57.14	**94.92**
ERTS-CASE_MOOD_PER	**73.16**	61.54	94.86
ERTS-DEF_GEN_NUM	72.6	**64.29**	**94.92**
ERTS-allFeat	72.91	59.26	94.78
FULL-noFeat	71.84	51.85	94.81
FULL-allFeat	72.52	57.14	94.67

Table 6: $F_{\beta=1}$ Results for ADJP, INTJP, and NP, across the different experimental conditions

As illustrated in Table 6, ERTS outperforms RTS and FULL in all corresponding conditions where they have similar corresponding morphological feature settings. ERTS-noFeat yields better results than RTS-noFeat and FULL-noFeat for the three different phrase types. The INTJP phrase is the worst performing of the three phrase types, however it marks the most significant change in performance depending on the experimental condition. We note that adding explicit morphological features to the base condition RTS yields consistently better results for the three phrase types. The highest performance for NP is yielded by ERTS-noFeat and ERTS-DEF_GEN_NUM with an $F_{\beta=1}$ of 94.92%.

The highest scores yielded for ADJP (73.16) and INTJP (64.29) are in an ERTS experimental condition. We also observe a slight drop in performance in NP for the ERTS conditions when the CASE MOOD and PER features are added. This might be due to the inconsistent or confusable assignment of these different features in the ATB.

5 Conclusions and Future Work

In this paper, we address the problem of Arabic base phrase chunking, BPC. In the process, we introduce a new enriched POS tag set, ERTS, that adds definiteness, gender and number information to nominals. We present an SVM approach to both the POS tagging with ERTS and the BPC tasks. The POS tagger yields 96.13% accuracy which is comparable to the results obtained on the standard reduced tag set RTS. On the BPC front, the results obtained for all conditions are significantly better than state of the art published results. This indicates that better linguistic tailoring of the *IOB* chunks creates more consistent data. Overall, we show that using the enriched POS tag set, ERTS, yields the best BPC performance. Even using ERTS with no explicit morphological features yields better results than using RTS in all conditions with or without explicit morphological features. These results are confirmed by closely observing specific phrases that are directly affected by the change in POS tag set namely, ADJP, INTJP and NP. Our results strongly suggests that choosing the POS tag set carefully has a significant impact on higher level syntactic processing.

6 Acknowledgements

This work was funded by DARPA Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of DARPA.

References

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.

Daniel Bikel. 2004. *Intricacies of Collins Parser*. Computational Linguistics.

Tim Buckwalter. 2002. *Buckwalter Arabic Morphological Analyzer Version 1.0*. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Catalog No.: LDC2002L49

David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. *Parsing Arabic Dialects*. Proceedings of the European Chapter of ACL (EACL).

Michael Collins. 2000. *Discriminative Reranking for Natural Language Parsing*. Proceedings of the 17th International Conference on Machine Learning.

Mona Diab, Kadri Hacioglu and Daniel Jurafsky. 2004. *Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks*. Proceedings of North American Association for Computational Linguistics.

Mona Diab, Kadri Hacioglu and Daniel Jurafsky. 2007. *Automated Methods for Processing Arabic Text: From Tokenization to Base Phrase Chunking*. Book Chapter. In Arabic Computational Morphology: Knowledge-based and Empirical Methods. Editors Antal van den Bosch and Abdelhadi Soudi. Kluwer/Springer Publications.

Nizar Habash and Owen Rambow. 2005. *Arabic Tokenization, Morphological Analysis, and Part-of-Speech Tagging in One Fell Swoop*. Proceedings of the Conference of American Association for Computational Linguistics (ACL).

Kadri Hacioglu and Wayne Ward. 2003. *Target word Detection and semantic role chunking using support vector machines*. HLT-NAACL.

Thorsten Joachims. 1998. *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Proc. of ECML-98, 10th European Conf. on Machine Learning.

Seth Kulick, Ryan Gabbard, and Mitch Marcus. 2006. *Parsing the Arabic Treebank: Analysis and Improvements. in* Treebanks and Linguistic Theories.

Taku Kudo and Yuji Matsumato. 2000. *Use of support vector learning for chunk identification*. Proc. of the 4th Conf. on Very Large Corpora, pages 142-144.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. *The Penn Arabic Treebank : Building a Large-Scale Annota ted Arabic Corpus*. NEMLAR Conference on Arabic Language Resources and Tools. pp. 102-109.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. *Text Chunking using transformational based learning*. Proc. of the 3rd ACL workshop on Very Large Corpora

Erik Tjong, Kim Sang, and Sabine Buchholz. 2000. *Introduction to the CoNLL-2000 shared task: Chunking*. Proc. of the 4th Conf. on Computational Natural Language Learning (CoNLL), Lisbon, Portugal, 2000, pp. 127-132.

Vladamir Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, USA.