A Statistical Method for Detecting the Arabic Empty Category

Hitham M. Abo Bakr
Computers & Systems Dept
Zagazig University
hithamab@yahoo.com

Khaled Shaalan
Faculty of Informatics
The British University in Dubai
khaled.shaalan@buid.ac.ae

Ibrahim Ziedan
Computers & Systems Dept.
Zagazig University
i.ziedan@yahoo.com

Abstract

In this paper we introduce a statistical approach for detecting the position of Empty-Category presented in Arabic Treebank. This can help in detecting the position of the elliptic personnel pronoun and overcoming, for some cases, the identification of dropped words within a sentence given the free word order nature of Arabic. The proposed approach requires a large corpus. The training for detecting the Empty-Category for each token is based on its Part Of Speech (POS), Base Phrase (BP)-chunk position, and the position of the token in the sentence. The Empty-Category detection is efficiently obtained using the Support Vector Machines (SVM) technique. We conducted an evaluation of the proposed diacritization algorithm, discussed the obtained results, and proposed various modifications for improving the performance of this approach.

Introduction

Major challenges in parsing Arabic text, include [1]: 1) The free word order nature of the Arabic sentence, and 2) The presence of an elliptic personal pronoun “Al-Damiir Al-Mustatir”. In [2] and [3], the authors provided a representation of the Empty-Category in the syntactic representation of the input Arabic sentence. Our objective is to propose a novel statistical methodology for detecting this Empty-Category and its position in the Arabic sentence. A significant advantage of this detection is that it is capable to resolve ambiguities such as determining the case ending diacritics. The methodology involves an Arabic Treebank as an important linguistic resource for training a tool that detects the Empty-Category.

The Arabic Treebank was created on top of a corpus that has already been annotated with POS tags. In our research, we took a decision to use ATB as it is a reliable linguistic resource used by various leading Arabic natural language research.

The main idea in our proposed methodology is to relate the Empty-Category of each token to: 1) its corresponding POS and chunk position, and 2) its position in the sentence.

In this research, the training is performed using Support Vector Machines (SVM) [5][6] technique that takes undiacritized sequence of tokens representing the input Arabic sentence and produces the training model that will be used for detecting the Empty-Category. YamCha is one of the state-of-the-art utility that widely and successfully used in natural language processing applications that adopt SVM technique. This has also led us to use YamCha [7][8] in our research. YamCha requires that the input is represented in a certain standard format properly suits its processing—we call it "YamCha input".

We have extracted from the ATB a sequence of tokens with its Part Of Speech (POS), Base Phrase (BP)-chunk and Empty-Category by using YamCha File Creator (YFC utility) that will produce the YamCha input. The basic approach used in YFC is inspired by the work of Sabine [9] for treebank-to-chunk conversion script, which we have enhanced in order to deal with the characteristics and peculiarities of Arabic language. This required adding some new features such as the Empty feature. The YamCha tool takes the output from the YFC to produce the training model.

The proposed approach

Treebanks are language resources that provide annotations of natural languages at various levels of structure: word level, phrase level, and sentence level.

Treebanks have become crucially important for the development of data-driven approaches to natural language processing.

The Arabic Treebank was created on top of a corpus that has already been annotated with POS tags. The Penn Arabic Treebank (ATB) began in the fall of 2001 [4] and has now completed four full releases of morphologically and syntactically annotated data: Version 1 of the ATB has three parts with different releases, some versions like Part 1 V3.0 and Part 2 V 2.0 are fully diacritized trees.

For example, the following diacritized sentence:

تَـلَـقَّـيـنـا الأَمْـر عِـنْـدَ الـسـاعَـة ٢٢:١٥ مِـن يَـوْم أَمْـس الْجُمْعَة بِقَصْف الـغـابَـة حَـيْـثُ كـان يُـوْجَـد,بِـحَـسَـب مَـعْـلُـومـاتـنـا,وَهِيَ الـتَـسْـمِـيَـة الَّـتِـي يُـطْـلِـقـهـا الـرُوْس فِـي إِشـارَة إِلَـى الْمُـقـاتِـلِـيْـنَ الـشِـيـشـان.

"talaq~aynA Al>amor Einoda AlsAEap 22:15 min yawom >amos AljumoEap biqaSof AlgAbap Hayovu kAn yuwojad, biHasab maEoluwmAtnA, luSuwS "wahiya Altasomiyap Al~atiy yuToliqhA Alruwos fiy <i$Arap <ilaY AlmuqAtiliyona AliyAn."

is represented in the ATB as:

(S (S (VP (VERB_PERFECT+PVSUFF_SUBJ:1P talaq~ay+nA) (NP-SBJ (-NONE- *)) (NP-OBJ (NP (DET+NOUN Al+>amor)) (PP-1 (-NONE- *ICH*))) (PP-TMP (PREP Einoda) (NP (NP (DET+NOUN+NSUFF_FEM_SG Al+sAE+ap) (NUM 22:15)) (PP (PREP min) (NP-TMP (NOUN yawom) (NP (NP (NOUN >amos)) (NP (DET+NOUN_PROP+NSUFF_FEM_SG Al+jumoE+ap))))))) (PP-1 (PREP bi-) (NP (NOUN -qaSof) (NP (NP (DET+NOUN+NSUFF_FEM_SG Al+gAb+ap)) (SBAR (WHADVP-3 (REL_ADV Hayovu)) (S (VP (VERB_PERFECT kAn) (VP (IV3MS+VERB_PASSIVE yu+wojad) (PUNC ,) (PP (PREP bi-) (NOUN -Hasab) (NP (NOUN+NSUFF_FEM_PL maEoluwm+At-) (POSS_PRON_1P -nA))) (PUNC ,) (NP-SBJ-2 (NOUN luSuwS)) (NP-OBJ-2 (-NONE- *)).

This representation is partially extracted from the tree file 20000715_AFP_ARB.0002.tree that is provided by the ATB Part 1 V.2. Figure 1 shows a graphical representation of this tree.

Empty categories can be identified in the Arabic Treebank. For example, in VSO the Arabic subject is analyzed under the VP, i.e. follows the verb. The subject (labeled NP-SBJ) occurs under the VP which comes after the verb, and is frequently a pro-drop (NP-SBJ *). In common VSO order, the lexical subject precedes the verb, which is labelled NP-TPC (topicalized) and traced to (NP-SBJ *T*) following the verb[1].

[1]http://www.ircs.upenn.edu/arabic/Jan03release/TBParsing-info.txt

Figure 2 uses the circle to highlight the Empty-Category in the graphical representation of the Treebank[2]. Empty-Category is indicated by one of the following tags: *, *T*, *ICH* and NO, see Table 1 for explanation of these tags.

Table 1 gives the meaning of these abbreviations.

Abbreviation	Meaning
*	Pro-drop subjects and passive traces
T	WH-traces, NP-TPC trace to subject
ICH	Rightward movement
NO	No Empty-Category in the next position

Table 1 Meaning of Abbreviations used in the Treebank

The idea is to relate the Empty-Category found next to each token with its POS and BP-chunk position as well as its position in the sentence. We made the training using Support Vector Machines (SVM) technique with undiacritized tokens because the expected input is undiacritized text.

We used YamCha File Creator (YFC utility)[3] to extract a sequence of tokens with its POS, BP-chunk and the Empty-Category found next to each token in the Treebank. The basic approach used in YFC is inspired by the work of Sabine [6] for treebank-to-chunk conversion script, which we have extended in order to be used with Arabic. This required adding some features like Empty-Category.

[2] Figure 1 and Figure 2 are the graphical representation for the Treebank files appeared using our Treebank Viewer tool, see http://www.staff.zu.edu.eg/hmabobakr/page.asp?id=53

[3] YFC utility is command line utility. We develop it using C++ to extract information from Penn Arabic Treebank ATB and create a Yamcha format to be used in the training process. http://www.staff.zu.edu.eg/hmabobakr/page.asp?id=53

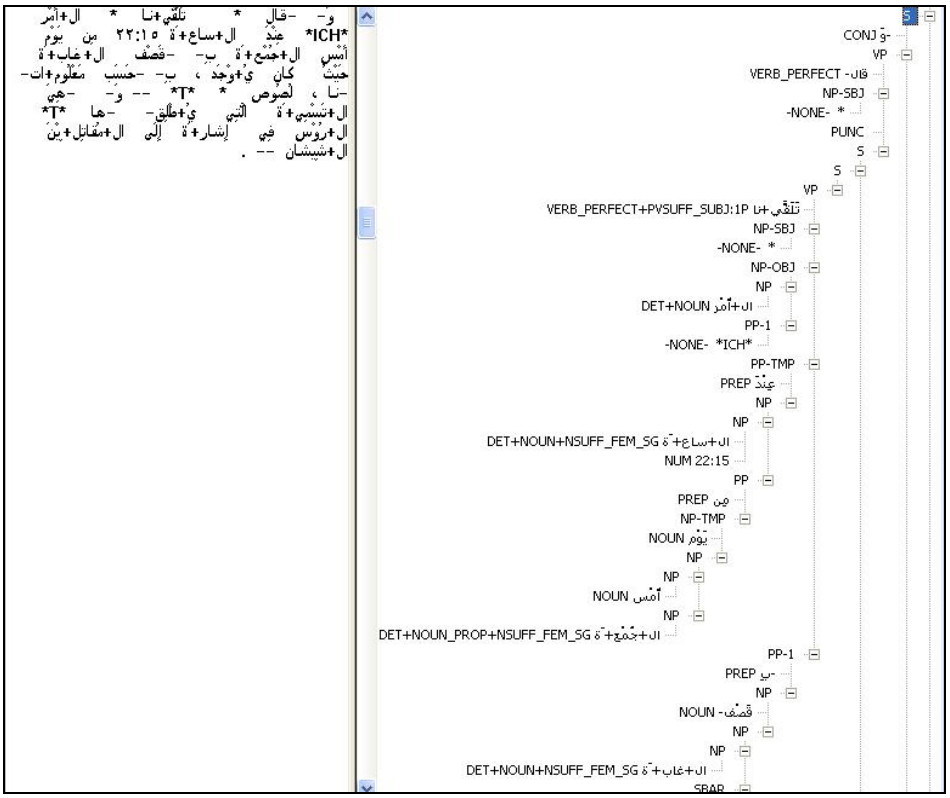

Figure 1: A graphical representation of an Arabic sentence extracted from the Penn Arabic Treebank

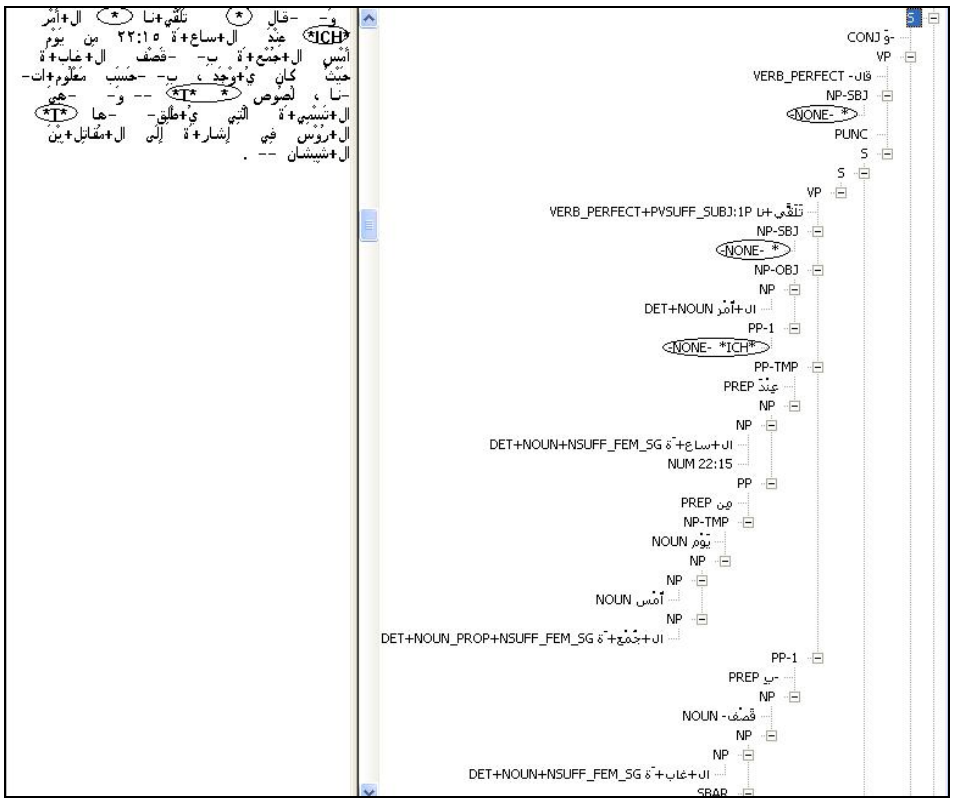

Figure 2: A graphical representation of an Arabic sentence extracted from the Penn Arabic Treebank

The output result from YFC utility for Empty-Category training process is shown in Table 3.

Token	POS	Chunk	Empty Flag
w	CC	O	NO
qAl	VBD	B-VP	*
"	PUNC	O	NO
tlqynA	VBD	B-VP	*
Al	DT	B-NP	NO
>mr	NN	I-NP	*T*
End	IN	B-PP	NO
Al	DT	B-NP	NO
sAEp	NN	I-NP	NO
22:15	CD	I-NP	NO
mn	IN	B-PP	NO

Table 2. Training file format for detecting Empty-Category

Evaluation

The proposed Empty-Category detection tool is trained and evaluated on the LDC's Arabic Treebank of diacritized news stories – Part 2 v2.0: catalog number LDC2004T02 and 1-58563-282-1. The corpus includes complete vocalization, i.e. diacritic marks are attached to all letters. We introduce here a clearly defined and replicable split of the corpus, so that the reproduction of the results or future investigations can accurately and correctly be established. This corpus includes 501 stories from the Ummah Arabic news text. There are a total of 144,199 words (counting non-Arabic tokens such as numbers and punctuation) in the entire 501 files – with one story per file. We divided the corpus into two sets: training data and the development/test (devtest) data. The devtest data are the files ended by character "7" like "UMAAH_UM.ARB_20020120-a.0007.tree" and its count was 38 files. The remaining 463 files were used for training about 90% of the total corpus. Hence, the devtest data represents about 10% of the total corpus. It is used to conduct an evaluation experiment that demonstrates the capability of our methodology in determining the Empty-Category and its position within an Arabic sentence. For determining the position of Empty-Category for Arabic, we used a standard SVM with a polynomial kernel, of degree 2 and C=1.0. Evaluation of the system is done by calculating the accuracy in detecting the Empty-Category.

Results

The results demonstrated that the accuracy for detecting the position of the Empty-Category of the system assuming gold tokenization, POS and BP-chunk has achieved 98.59% of accuracy.

Class Name	Precision	Recall	F-Measure	Correct
NO	0.991	0.994	0.993	15138
T	0.881	0.855	0.868	614
0	0.830	0.672	0.743	131

Table 3. Some Detail results for some categories

Table 3 presents some detailed results for some Empty-Categories. The accuracy of results shows a perfect results but actually because the high accuracy of detecting the "NO class". The Empty-Category detection show acceptable results, but from 80% and 85%, because the number of Empty-Category position in the test sample was very rare (614 and 131) but the "NO class" was about 15138.

Conclusion

This paper addresses the problem of detecting the Empty-Category that is important when parsing the Arabic sentence. A statistical approach is proposed for detecting the position of Empty-Category. The Arabic Treebank is used for training and testing the implementation of this approach. The evaluation results demonstrated the capability of the approach in detecting Empty-Category such as the elliptic personnel pronoun and dropped words.

References

[1] Othman E., Shaalan K. and Rafea A., (2004), "Towards Resolving Ambiguity in Understanding Arabic Sentense". In NEMLAR Conference on Arabic Language Resources and Tools, Cairo, Egypt.

[2] Mohammad, Mohammad. (1999), "Word Order, Agreement and Pronominalization in Standard and Palestinian Arabic: Current Issues in Linguistic Theory", vol.181. Amsterdam: John Benjamins.

[3] Chalabi A. (2004), "Elliptic Personal Pronoun and MT in Arabic", JEP-TALN 2004, Arabic Language Processing Fez, 19-22 April 2004.

[4] Maamouri M., Bies A., and Buckwalter T., (2004), "The Penn Arabic Treebank: Building a large-scale annotated arabic corpus". In NEMLAR Conference on Arabic Language Resources and Tools, Cairo, Egypt.

[5] Cristianini N. and Taylor J.S., (2000), "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods", The Press Syndicate of the University of Cambridge, Cambridge, United Kingdom.

[6] Hearst M. A., (1998), "Support Vector Machines," IEEE Intelligent Systems, vol. 13, no. 4, pp. 18-28, Jul/Aug, 1998.

[7] Kudo T. and Matsumoto Y., (2003), " Fast methods for kernel-based text analysis," In Proceedings of the 41st Annual Meeting on Association For Computational Linguistics - Volume 1 (Sapporo, Japan, July 07 - 12, 2003). Annual Meeting of the ACL. Association for Computational Linguistics, Morristown.
[8] Kudoh T.and Matsumoto Y., (2000), "Use of Support Vector Learning for Chunk Identification," In Proceedings of the 4th Conference on CoNLL-2000 and LLL-2000, pages 142-144.
[9] Sang E. and Buchholz S., (2000),”Introduction to the CoNLL-2000 Shared Task: Chunking”, Proceeding of CoNLL-2000 and LLL-2000, Page 127-132, Lisbon, Portugal.