COMBINATORIAL CLASSIFICATION FOR CHUNKING ARABIC TEXTS

Fériel Ben Fraj[1] and Maroua Kessentini[2]

RIADI Laboratory, Manouba University, Manouba, Tunisia
[1]Feriel.BenFraj@riadi.rnu.tn
[2]maroua.kessentini@gmail.com

ABSTRACT

Text parsing has always benefited from special attention since the first applications of natural language processing (NLP). The problem gets worse for the Arabic language because of its specific features that make it quite different and even more ambiguous than other natural languages when processed. In this paper, we discuss a new approach for chunking Arabic texts based on a combinatorial classification process. It is a modular chunker that identifies the chunk heads using a combinatorial binary classification before recognizing their types based on the parts-of-speech of the chunk heads, already identified. For the experimentation, we use over than 2300 words as training data. The evaluation of the chunker consists of two steps and gives results that we consider very satisfactory (average accuracy of 89,60% for the classification step and 80,46% for the full chunking process).

KEYWORDS

Classification, chunking, combinatorial system, Arabic language.

1. INTRODUCTION

Parsing is a central task in the NLP applications and tasks. It can be either shallow parsing or deep parsing. These two types do not need the same information amount and do not give the same result. Therefore, shallow parsing (also called skimming) consists of splitting sentences into chunks (or phrases), and does not seek how the phrases attach to each other. Chunks are words (or sets of words) that compose correctly formed syntactic sub-structures. It consists of a head word and its neighboring modifier words (such as adjectives, adverbs and/or modifiers). However, deep parsing should return complete syntactic structures. Also, chunking must be faster and more robust than deep parsing.

Chunking Arabic texts needs special attention given that the Arabic language is more ambiguous than other natural languages (especially the Indo-European languages). Indeed, Arabic has several characteristics among which we mention the following:

- Vocalic ambiguity: the oversight of the vocalic marks increases the ambiguity of words' comprehension;
- Grammatical ambiguity: several words may have the same grammatical interpretation;
- Agglutination of the clitics to the simple textual forms;
- Problems related to the segmentation of texts into sentences;

DOI : 10.5121/ijaia.2012.3506

- Abundance of subordinate structures;
- Elliptic and anaphoric structures.

Two main approaches are used to deal with the chunking problem: namely, the rule-based methods and the learning-based methods. While the first category involves a set of grammatical rules, the second requires real labelled training data.

The remainder of this paper is divided into four main parts. Section 2 discusses the state of the art related to the chunking problem. We consider both the Arabic chunking and the chunking based on classification. This study induces us to choose a learning-based approach. Next, section 3 illustrates our combinatorial approach for chunking Arabic texts into chunks. The chunks can be one of the three following types: NP (Nominal Phrase), VP (Verbal Phrase) and PP (Prepositional Phrase). Then, section 4 explains the technique we used to evaluate the efficiency of the developed chunker and the obtained results. Ultimately, section 5 concludes the paper and presents some future improvements.

2. RELATED WORKS

The first chunker was developed by Abney [1] who has noted that when we read a text, we read it a chunk at a time. Later, several chunkers for different languages, including the Arabic language, have been created. In this section, we discuss the previous works related to the research questions addressed in this paper, mainly researches dealing with Arabic text chunking and those making use of the classification for solving the same problem.

2.1. Arabic Chunking

The state of the art of the previous Arabic chunking researches leads us to construct the following table that presents a comparative study between a set of Arabic chunkers. The comparison is done through the following criteria:

- The approach type: rule-based or corpus-based approach;
- The input type that is to say if the input of the chunker is voweled and labeled or not;
- The size of the test data;
- Performance as precision, recall or error score.

Table 1. Comparative study of Arabic chunkers.

Chunker	**Research team**	**Approach**		**Input type**	**Test Corpus**	**Performances**
		Rule-based	**Corpus-based**			
Belghith [2]	Belghith (Laris-Miracl Labratory)	×		labelled	**??**	**??**
Baloul [3]	Baloul and De Mareuil (Computer Sciences Laboratory University of Maine-France)	× (grammar of chunks)		labelled	Multext Corpus	Error score = 7%
TAGGAR [4]	Zemirli and Khabet (National Institute of Computer Sciences (Algeria))	×		Voweled and labelled	Corpus of 5563 words	Precision = 98%
ASVM [5]	Diab and al. (University of Columbia)		× (PATB : Penn Arabic TreeBank)	labelled	400 sentences from PATB	precision = 92,06% recall = 92,09%
Mohamed [6]	Mohamed and Omar (Faculty of Information Science and Technology, Malaysia)	×		labelled	70 sentences (1776 words)	f-score = 97%

All the works mentioned in this table, except ASVM [5], are rule-based chunkers. However, we consider that the construction of grammars that cover all the syntactic structures is a difficult task to achieve, especially for Arabic that has several exceptional grammatical structures increased by the characteristics mentioned above.

Furthermore, some of chunkers are specially developed to be used in other linguistic applications as the syntactic error correction [2] or the speech synthesis as for the works of Baloul [3] and Zemirli [4]. On the other hand, a chunker must be well designed in order to be used in other research fields as information retrieval.

2.2. Chunking by Classification

The classification has been used as solution for the chunking problem in some researches as that of Diab [5], mentioned in the previous subsection. Diab processes the Arabic language. Furthermore, there are other works [7] and [8] that use the multi-class classification for the identical problem. For the first one [7], the researchers employed a generalization of the original Winnow method. They used some discriminative features, especially parts-of-speech. They have defined eleven chunk types. Each of these types is represented by one of the three labels:

- B-X: first word of a chunk of type X;
- I-X: non-initial word in a chunk of type X;
- O: word outside of any chunk.

For the second one, the Kashmiri is the target language. Bhat and Sharma [8] divide a multi-class classification problem into binary classification problem. So, the chunking process is broken down into two stages. The first stage uses the conditional random fields (CRFs) in order to identify the chunks boundaries. The second one assigns each chunk its appropriate chunk type. The overall system is performed at an accuracy of 94.85%.

3. Approach Presentation

We consider that the division of the chunking task can reduce the problem complexity and improve the chunking efficiency. In addition, we assume that using a learning approach is more beneficial than using a rule-based approach since the construction of a grammar that covers all the Arabic syntactic structures (exceptional and specific) is very hard or even impossible. Hence, we choose to consider a modular approach which is based on two steps:

- The identification of the chunk heads using a combinatorial classification;
- The recognition of the chunk types based on the parts-of-speech of the identified chunk heads.

Figure 1 presents the architecture of this modular approach.

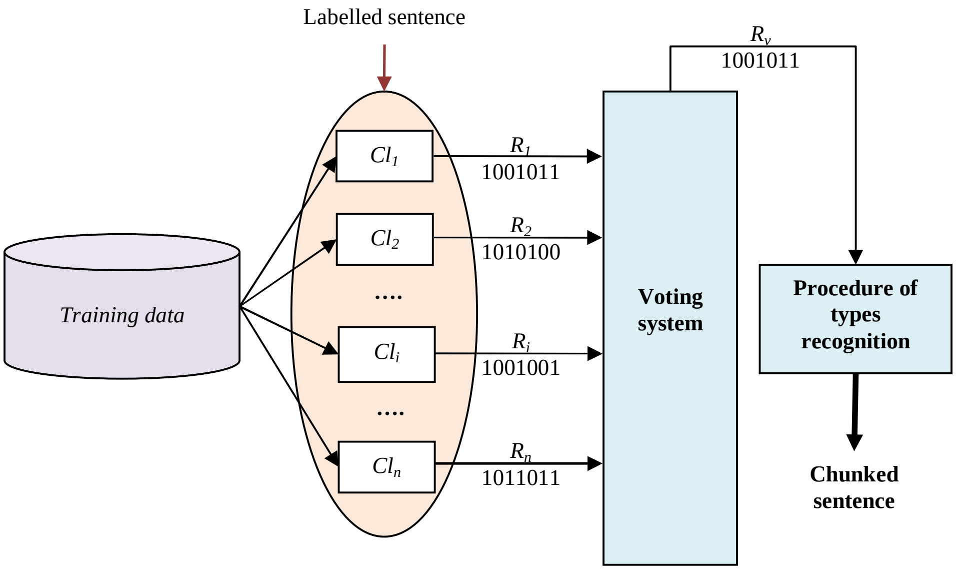

Figure 1. Architecture approach

3.1. Identification of the Chunk Heads

In order to identify the chunk heads, we use several classifiers. Each classifier receives the labelled sentence word by word and should decide if the target word is a chunk head or not. So, it is a binary classification. We choose the binary classification in order to improve the chunking efficiency. In fact, when using a binary classification, the error score is of $1/2$ for each classification step. In contrast, when using n classes, the error score is estimated $\frac{n-1}{n}$. In addition, we assume that binary classification can speed up the identification of the head chunks.

In order to disambiguate the two classes, we use some discriminative attributes which are various. We use several morphosyntactic information [9] related to the word that will be classified and its surrounding neighbors in a window of [-1, 1]. So, we mention the following characteristics:

- The stem part-of-speech;
- The parts-of-speech of the proclitic and enclitic if there exist;
- The gender and the number if the word is a nominal item;
- The transitivity and the pronoun if the word is a verb;
- The morphosyntactic information related to the previous word and the following one.

Afterward, each classifier should return a binary sequence. The sequence's length is equal to the sentence's length. Every digit indicates if the corresponding word is a beginning of a chunk (1) or not (0). Different results are gathered and a voting system is used to decide the most likely sequence. This voting system assigns the word the class elected by the majority of the singular classifiers. Thus, it fulfils the following hypothesis:

$$R_v = \begin{cases} 1 \text{ if } & \sum_{i \le n} R_i > \frac{n}{2} \\ 0 \text{ otherwise} \end{cases}$$

If we consider that the number of the used classifiers is n and that the sum of the resulting classes is greater than $n/2$, then the result (R_v) of the voting system is 1. However, R_v is equal to 0. If the number of classifiers that voted 1 and the number of the classifiers that voted 0 is the same, then the result of the best performing classifier will be elected.

Example: Let us consider a sentence of 10 words and we choose to combine 3 classifiers.

Cl_1 result:	1001010011
Cl_2 result:	1010100101
Cl_3 result:	1010010010
Voting result:	1010010011

3.2. Recognition of the Chunk Types

After identifying the heads of the chunks, we provide a procedure to recognize the types of these chunks. For this purpose, we step in the parts-of-speech and proclitics (if there exist) of the identified chunk heads. The procedure is recursive. Indeed, in a first stage, it checks whether the chunk head is a verb, a nominal entity or a preposition:

- If it is a verb, then the chunk type is a Verbal Phrase (VP) مركّب فعلي;
- If it is a preposition, then the chunk type is Preposition Phrase (PP) مركب جر;
- If it is a nominal, then we should verify if the word is agglutinated to a proclitic and this proclitic is preposition or not, if it is, then the chunk is PP, else it is a Nominal Phrase (NP) مركب اسمي ;

If the chunk head is a non-significant word (modifier for example), the recursion should pass to the following word if it is not classified as a chunk head.

4. EXPERIMENTATION AND RESULTS

We exploit the validity of our choices, especially the classification process and the classifiers' combination, through several tests. Thus, we test the classification process. After that, we inspect the full chunking process.

4.1. Classification Evaluation

The classification constitutes the most important stage of our chunker that is why we have done some experiments to evaluate its impact in the whole solution. We have chosen five classifiers from the java API weka. The choice covers different algorithms of different classifiers types; namely: the rule-based, the probabilistic, the case-based, the functions and the decision trees methods. The evaluation is made with a set of data of over 2375 words. It consists of 1424 chunks that compose 283 sentences. The corpus is a set of literary texts. In order to evaluate this chunking step, we estimate the precision that is equal to the number of the correctly classified words divided by the number of the words in the test set. The results are listed in the table below.

Table 2. Evaluation of the classification step.

	Probabilistic (NaiveBayes)	**Decision tree (J48)**	**Function (SMO)**	**Empiric (KStar)**	**Rul-based (decision table)**	**Voting system**
Test1	74,17	93,12	72,98	83,46	91,70	89,81
Test2	77,82	92,19	76,38	89,11	88,09	92,60
Test3	72,53	87,98	70,81	79,61	90,34	84,97
Test4	77,36	75,00	71,31	84,21	61,31	90,52
Test5	73,33	92,10	74,73	85,52	88,42	90,11
Average	**75,04**	**88,07**	**73,24**	**84,38**	**83,97**	**89,60**

The evaluation of the classification step consists of 5 tests. For each one, we consider 80% of the corpus for the training data and 20% for the test data. We notice that the performance average of the voting system (89,60%) is better than the performances of the different singular classifiers. By cons, the performances of the singular classifiers are quite various and there is not a classifier that always gives the best performance. The results are satisfactory due to the small size of the training data. Increasing the size of the corpus by other texts will improve the efficiency of the different individual classifiers and, therefore, that of the voting system.

For the previous tests, we have chosen the classifiers arbitrarily. For the following set of tests, we chose using classifiers sorted according to their respective performances. Then, we proceeded by combining the two worst performing classifiers and at each phase we added a classifier of higher performance to the combination. We have constructed the two following graphs (Figure 2). We notice that the performance of the combination (vote) increases gradually with the number and the performance of the individual classifiers. This performance gain is facing a loss in response time of the voting system.

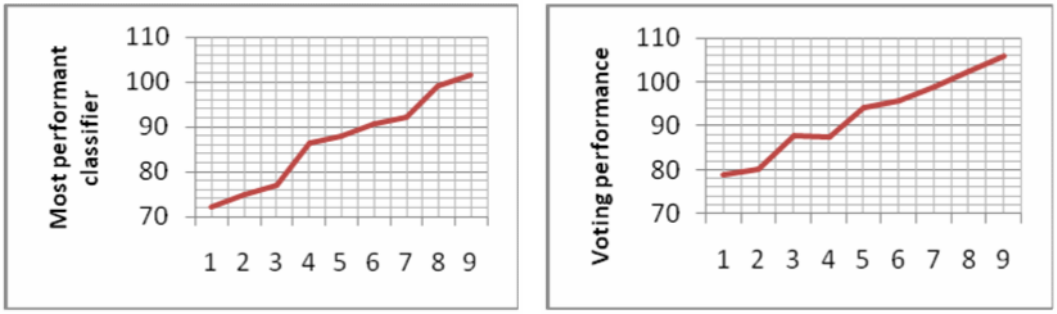

Figure 2. Influence of the classifiers' number on the global combinatorial classification performance

4.2. Full Chunking

Our approach does not stop at the classification stage since it contains a recognition procedure that specifies the chunk types. For this evaluation step, we use the accuracy rate calculated as follows:

$$\text{Chunking accuracy} = \frac{\text{number of correctly indentified chunks}}{\text{total number of chunks}}$$

So, the results of the full chunking tests are presented in the following table.

Table 2. Evaluation of the full chunking.

	Percentage of correct recognitions
Test1	80,81%
Test2	84,53%
Test3	84,70%
Test4	74,24%
Test5	78,03%
Average	80,46%

For these experiments, we use the same set of texts as for the classification tests. The chunker affects the types to the identified heads of chunks. The recognition of the VP chunks is better than the recognition of the NP and PP types. The recognition error rate is equal to 5,44%, since the recognition procedure is only based on the parts-of-speech. The PP is the least recognized type (error rate equal to 27,47%) as the prepositions can be proclitics which makes the recognition ambiguous. Thus, we can improve the procedure recognition by adding more syntactic information. For the NP type, the error rate is equal to 23, 48% due to the disparity of the Arabic

NP structures (NP of annexation (مركب إضافي), adjectival NP (مركب نعتي), approvingly NP (مركب بدلي), corroborative NP (مركب توكيدي), quasi-propositional NP (مركب شبه إسنادي), etc.). Our chunker can parse the sentence above as follows:

في ذات يوم(PP)# اجتاحت(VP)# الحارة(NP)# معركةشاملة(NP).

(*In one day, an overall battle invades the street*)

Our recognition procedure can be improved to identify these different syntactic structures. Furthermore, it is worth mentioning that the chunking results are satisfactory (80,46% in average) according to the limited size of the training data used at the classification step. The errors of the classification step persist when recognizing the chunk types. We expect to improve our parser so that the two stages (classification and recognition) cooperate with each other to provide better results.

3. Conclusion

Focusing on the previous Arabic chunking works, most of them use rule-based approaches, whereas learning-based methods make use of real information that can cover different exceptional syntactic structures. In this paper, we presented an Arabic chunking approach that proceeds in two steps. Firstly, a combinatorial binary classification stage points out the chunk heads. Secondly, a recognition stage identifies the types of the chunks recognized by their heads. After the experimental tests, we consider the results very satisfactory in spite of the lack of the training data. These results can be improved if we enlarge the size of the data used for training. In addition, the cooperation between the two steps can enhance the performances. The output of our chunker will be the input for a deep Arabic parser.

References

[1] Abney, S. (1991). Principle-based parsing: computation and psycholinguistics.1st Edn., Springer, Dordrecht, ISBN: 0792311736, pp: 408.

[2] Hadrich-Belghith, L. (1999). Traitement des erreurs d’accord de l’arabe basé sur une analyse syntagmatique étendue pour la vérification et une analyse multicritère pour la correction, PhD thesis, Faculty des Economic Sciences and Management of Sfax, Tunisie.

[3] Baloul, S. and De Mareuil, B. (2002). Un modèle syntactico-prosodique pour la synthèse de la parole à partir du texte en arabe standard voyellé. In Karttunen, L. (1974). Presupposition and linguistic context. Theoretical Linguistics, 1 pp. 181-94. Also in Pragmatics : Reader, A., and Davis, S (eds), pages 406–415.

[4] Zemirli, Z. and Khabet, S. (2004). TAGGAR : Un analyseur syntaxique dédié à la synthèse vocale de textes arabes voyellés, In Actes des Journées d'Etudes sur la Parole (JEP) et Traitement Automatique des Langues Naturelles (TALN), Fès.

[5] M. Diab, K. Hacioglu, and Jurafsky, D. (2004). Automatic tagging of arabic text : from raw text to base phrase chunks. In 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04), pages 149–152.

[6] Mohammed ,M-A. and Omar, N. (2011). Rule based shallow parser for Arabic language. Journal of Computer Sciences. 7(10): 1505–1514.

[7] Zhang, T., Damerau, F. and Johnson, D. (2002). Text Chunking based on a Generalization of Winnow, Journal of Machine Learning Research 2. Submitted 9/01; Published 3/02, pages 615-637.

[8] Bhat, R. A. and Sharma, D. M. (2011). A hybrid approach to kashmiri shallow parsing, In actes de The 5th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC-2011).

[9] Ben Othmane Zribi C., De la synthèse lexicographique à la détection et à la correction des graphies fautives arabes, PhD Thesis, Paris XI University, Orsay, 1998.

Authors

Fériel Ben Fraj received her PhD in computer sciences in 2010 from Manouba University,Tunisia. Currently she is an associate professor at the High Institute of Applied Sciences and Technology of Gabes in Tunisia. She is actually a member of RIADI Research Laboratory at University of Manouba. Her current research interests are in the area of Arabic language processing. Her recent works focus on Arabic parsing based on machine-learning approaches, construction of formal grammars and Arabic Treebank.

Maroua Kessentini received her Engineering Diploma in Computer Sciences from National School of Computer Sciences, University of Manouba, in June 2012. Currently she is a master's degree student at the RIADI research laboratory, University of Manouba.