Modeling Arabic Language using statistical methods

Karima Meftouh, Med Tayeb Tayeb Laskri, Kamel Smaïli

MODELING ARABIC LANGUAGE USING STATISTICAL METHODS

Karima Meftouh* and M. Tayeb Laskri

Badji Mokhtar University, Computer Science Department, BP 12 23000 Annaba, Algeria

and Kamel Smaili

INRIA-LORIA, Parole Team, BP 101 54602 Villers, Les Nancy, France

الخلاصة:

نقترح في هذا المقال دراسة النماذج الإحصائية للغة العربية. أولا : سوف نقوم بإجراء العديد من التجارب باستخدام تقنيات ملمس مختلفة على مدونة صغيرة الحجم والمستخرجة من صحيفة يومية. قلة البيانات تقودنا للجوء إلى حلول أخرى دون زيادة حجم المدونة. وقد استخدمنا تقنية تقطيع الكلمات من أجل زيادة القابلية الإحصائية للمدونة مما أدى إلى نتائج أفضل. أما التجربة الثانية فتتمثل في دراسة سلوك نماذج إحصائية تقوم على أنواع مختلفة من المدونات. كذلك سنبين أن استعمال النماذج البعيدة يحسّن الأنموذج الأساسي. أخيرا نقترح دراسة مقارنة لنماذج إحصائية للغة العربية ولغات أجنبية عدة. الهدف من هذه الدراسة هو فهم كيفية تحسين كل أنموذج من هذه اللغات. وبالنسبة للغة العربية ، فإن النماذج المحسوبة بتقنية «ويتن- بال» هي الأكثر كفاءة.

*Corresponding Author:

Tel: 00213 775192905

Fax: 00213 38831618

E-mail: karima.meftouh@univ-annaba.org

Paper Received July 7, 2010; Paper Revised October 25, 2010; Paper Accepted October 30, 2010

ABSTRACT

In this paper, we propose to investigate statistical language models for Arabic. First, several experiments using different smoothing techniques are carried out on a small corpus extracted from a daily newspaper. The sparseness of the data leads us to investigate other solutions without increasing the size of the corpus. A word segmentation technique has been employed in order to increase the statistical viability of the corpus. An n-morpheme model has been developed which leads to a better performance in terms of normalized perplexity. The second experiment concerns the study of the behavior of statistical models based on different kinds of corpora. The introduction of a distant n-gram improves the baseline model. Finally, we propose a comparative study of statistical language models for Arabic and several foreign languages. The objective of this study is to understand how to better model each of these languages. For foreign languages, trigram models are most appropriate whatever the smoothing technique used. For Arabic, the n-gram models of higher order smoothed with the Witten-Bell method are more efficient.

Key words: language model, morphemes, perplexity, segmentation, smoothing techniques, corpora

MODELING ARABIC LANGUAGE USING STATISTICAL METHODS

1. INTRODUCTION

Statistical techniques have been widely used in automatic speech recognition and machine translation over the last two decades [1]. This success has been obtained essentially for the so-called "resource rich languages", such as English, French, and Chinese. Since 2001, the Arabic language has become a priority for several researchers throughout the world.

Arabic has a rich morphology characterized by a high degree of affixation, interspersed vowel patterns, and roots in word stems, as shown in Section 2. As in other morphologically rich languages, the large number of possible word forms entails problems for robust language model estimation.

A statistical language model is used to build up sequences of words, classes, or phrases which are considered linguistically valid in accordance with a corpus without any use of external knowledge. For each linguistic event, a probability is estimated to indicate its likelihood. An event is any potential succession of words.

The common model used in the literature is the well-known n-gram. A word is estimated in accordance to the (n-1) previous words. To be efficient, this model needs a huge amount of data to train all the required parameters. The necessary resources for this language are not as important as what we have for the Indo-European languages.

In the present work, we investigate several classical statistical language models in order to study their pertinence for Arabic [2]. The sparseness data compels us to test several smoothing techniques in order to find out the best model. The next experiment concerns the study of the behavior of statistical models based on different kinds of corpora. The introduction of distant n-gram improves the baseline model. We also conduct a comparative study of Arabic n-gram model performances and those of several other languages. We first compare Arabic and French n-gram models [3]. Then we extend this comparison to other languages: English, Portuguese, and Greek (from the Indo-European languages family), and Finnish (which belongs to another family), in order to check if the made observations remain valid for languages belonging to the same family. To our knowledge, this kind of study has never been done and we would like to investigate the differences between these languages over their respective n-gram models.

2. AN OVERVIEW OF ARABIC

Arabic, one of the six official languages of the United Nations, is the mother tongue of 300 million people. Unlike Latin-based alphabets, the orientation of writing in Arabic is from right to left. The Arabic alphabet consists of 28 letters and can be extended to ninety by additional shapes, marks, and vowels. Each letter can appear in up to four different shapes, depending on whether it occurs at the beginning, in the middle, at the end of a word, or alone. Table 1 shows an example of the letter ف / "f" in its various forms. Letters are mostly connected and there is no capitalization.

Table 1. The Letter *"f"* / ف in Its Various Forms

Isolated	Beginning	Middle	End
ف	فـ	ـفـ	ـف

Arabic is a Semitic language. The grammatical system of the Arabic language is based on a root-and-pattern structure and Arabic is considered a root-based language with no more than 10000 roots and 900 patterns [4]. The root is the bare verb form. It is commonly three or four letters and, rarely, five. Pattern can be thought of as a template adhering to well-known rules.

Arabic words are divided into nouns, verbs, and particles. Nouns and verbs are obtained from roots by applying templates to the roots in order to generate stems and then by introducing prefixes and suffixes [5]. Table 2 lists some templates (patterns) to generate stems from roots. The examples given below are based on the root درس /"drs".

Table 2. Some Templates to Generate Stems from the Root درس "drs": C Indicates a Consonant, A a Vowel

Template	Stem
فعل CCC	درس drs study
فاعل CACC	دارس dArs student
مفعــول mCCwC	مدروس mdrws studied

Words are compact, which means that a word can correspond to an entire phrase. That is why some prefixes and suffixes correspond to whole words in other languages. In Table 3, we present the different components of a single word وكررتها which corresponds to the phrase "and she repeated it".

Table 3. An Example of an Arabic Word

Arabic	English
و	And
كرر	Repeated
ت	she
ها	it

3. N-GRAM MODELS

The goal of a language model is to determine the probability $P\left(w_1^k\right)$ of a word sequence w_1^k. This probability is decomposed as follows [6]:

$$P(w_1^k) = \prod_{i=1}^{k} P(w_i / w_1^{i-1}) \tag{1}$$

The most widely used language models are n-gram models [6]. In n-gram language models, we condition the probability of a word on the identity of the last (n-1) words.

$$P(w_1^k) = \prod_{i=1}^{k} P(w_i / w_{i-n+1}^{i-1}) \tag{2}$$

The choice of n is based on a trade-off between detail and reliability, and will be dependent on the available quantity of training data [7].

Because of the sparseness data, in statistical language models, parameters have to be smoothed. The objective is to fine-tune probabilities to overcome the problem of missing data. Several methods exist in the literature. We can cite Good-Turing [8], Witten-Bell [9], Linear [10], Kneser-Ney [11].

The quality of a language model is estimated either by entropy or by perplexity. The perplexity is inspired from entropy and is given by the following formula [12]:

$$PP = \frac{1}{\sqrt[n]{P(w_1, ..., w_n)}} \tag{3}$$

4. ARABIC N-GRAM MODELS

In this section, we report the results we obtained with Arabic statistical modeling. We first describe the used data.

4.1. Data Description

The experiments reported in this section were conducted on corpora extracted from Al-Khabar (an Algerian daily newspaper). Al-Khabar is written in modern standard Arabic, the one used by all the official media in the Arabic world. One of the specificities of the Arabic language is that a text can be read without using any vowel. That is why articles in newspapers are written without diacritics. The corpora we used contained 80K words for training and 5K words for test. Figure 1 shows a sample of the training corpus.

سوق اهراس
انجراف التربة يستهلك الملايير دون جدوى
عجزت المصالح التقنية لولاية سوق اهراس عن مجابهة مشكل انجراف التربة الذي يلحق أضرارا كبيرة بشتى القطاعات وتنجر عنه خسائر مالية هامة تكون على حساب متطلبات تنموية أخرى وهو ما جعل الولاية من أكثر الولايات تخلفا وأكثرها تأخرا في المرافق الإجتماعية الضرورية

Figure 1: A sample of the training corpus

For both the following experiments, the language models have been smoothed by three techniques: Good-Turing, Witten-Bell, and linear.

4.2. Word-Based n-Gram Models

The baseline models (bigram, trigram, and quadrigram) are calculated with a vocabulary of the most frequent 2000 words. Any word not in the vocabulary is replaced in the corpora by an abstract entity noted UNK (UNKnown word). The UNK may distort the interpretation of results because of its occurrence [13], so it can act in favour of a better language model within the meaning of perplexity if the vocabulary has a weak cover. Table 4 and Table 5 show the performance in terms of test perplexity without and with UNK, respectively. The rate of unknown words is 30.19%.

Table 4. Perplexity and Entropy Performance Without UNK

n	Good-Turing		Witten-Bell		Linear	
	Perplexity	Entropy	Perplexity	Entropy	Perplexity	Entropy
2	289.10	8.18	267.86	8.07	309.29	8.27
3	292.36	8.19	278.87	8.12	321.50	8.33
4	307.51	8.26	311.97	8.29	335.14	8.39

Table 5. Perplexity and Entropy Performance Including UNK

n	Good-Turing		Witten-Bell		Linear	
	Perplexity	Entropy	Perplexity	Entropy	Perplexity	Entropy
2	76.66	6.26	76.03	6.25	82.92	6.37
3	81.55	6.35	81.18	6.35	92.09	6.52
4	88.07	6.46	89.25	6.48	97.67	6.61

Indeed, the more the corpus contains UNK, the greater the probability that this fictive word is large, which leads to less perplexity. It is, thus, desirable to always calculate perplexity without UNK. Note also that the values of perplexity without UNK are high and increase according to the order of the model. This is due to the weak size of the training corpus. To take into account the sparseness data issue, we propose to split words into morphemes. This operation leads to an increase in the frequency of units and, consequently, to a reduction in the percentage of unknown words.

4.3. Morpheme-Based n-Gram Models

Languages with rich morphology generate so many representations from the same root. Often, this makes them highly flexional and, consequently, the perplexity could be high [14]. An Arabic word consists of a sequence of morphemes respecting the following pattern: prefix*-stem-suffix* (* denotes zero or more occurrences of a morpheme). We define an n-morpheme model as an n-gram of morphemes. In this case, the corpus is rewritten in terms of morphemes, as in the example of Figure 2.

سوق_اهراس
انجراف ال تربة يستهلك ال ملايير دون جدوى
عجزت ال مصالح ال تقنية لـ ولاية سوق_اهراس عن مجابهة مشكل انجراف ال تربة ال ذي يلحق أضرارا كبيرة بـ شتى ال قطاعات و تنجر عنه خسائر مالية هامة تكون على حساب متطلبات تنموية أخرى و هو ما جعل ال ولاية من أكثر ال ولايات تخلفا و أكثرها تأخرا في ال مرافق ال إجتماعية ال ضرورية

Figure 2: A sample of Arabic morphemes corpus

When we proceed to a decomposition of words into prefix-stem-suffix, we modify the number of items constituting the original corpus W. To make the comparison of the two models relevant, the perplexity has to be normalized [15], which leads to the following new formula of perplexity:

$$PP_n = 2^{\frac{n_1}{n_2} log_2(PP)} \quad (4)$$

where n_1, n_2 correspond, respectively, to the size of the original corpus and the rewritten one.

In this work, we are only interested in the segmentation of prefixes. We developed a semi-automatic tool that allowed us to perform this segmentation. Table 6 lists the frequently used prefixes in Arabic. Examples of Arabic words and their segmentation are given in Table 6.1.

Table 6. Prefixes and Their Meanings

و	And	لـ	To
كـ	Like	بـ	With
فـ	Then	الـ	The

Table 6.1. Examples of Words and Their Prefixes Segmentation

Word before segmentation	prefixes	Word after segmentation
الولايات	الـ	ولايات
وليقوم	لـ, و	يقوم

To make the corpus statistically reliable and to fit the reality of the Arabic language, some words have been gathered. That is why, for instance, we concatenate the town's name composed of two or more words. For instance, the city **سوق أهراس** is rewritten as **سوق_أهراس**.

This operation is handled by using a predefined list of composed words. Work is under way to find out how to automatically sequence Arabic words [16].

The transformation of the initial corpora leads to a training and a test corpus of 110K and 6.9K tokens, respectively. Table 7 illustrates the n-morpheme model.

Table 7. Normalized Perplexity's Values Without UNK

n	Good-Turing	Witten-Bell	Linear
2	173.52	170.18	188.05
3	130.05	123.55	145.61
4	133.06	126.23	146.93

These results show an improvement of 55.7% in terms of 3-gram perplexity using the Witten-Bell smoothing technique. We can state that for a small corpus, the segmentation of words improves the language model and this, whatever, used the technique of smoothing.

5. INFLUENCE OF THE CORPUS NATURE ON THE QUALITY OF N-GRAM MODELS

In order to study the influence of the data type on the quality of language models, we used two different corpora. The first (Corp1) is made up of articles of the Algerian newspaper Al-Khabar, used in the previous experiments. The second (Corp2) is extracted from the *CAC corpus* compiled by Latifa Al-Sulaiti within her thesis framework [17]. Texts constituting this corpus were collected from three main sources: magazines, newspapers, and web sites. Figure 3 shows a sample of the training corpus Corp2.

امرأة وثلاث قطط ودار مظلمة لا يعرف لها النور منفذا ولا طريقا الشمس لم تدخلها قط والهواء نفس الهواء والرطوبة القاتلة والأرض التي تنشق والسقف الذي انهار والجدران
امرأة الوحشة تلفها والوحدة تضيق عليها و القطط طوافات حولها

Figure 3: A sample of the training corpus Corp2

To compare the performance of models computed on these corpora, we had to take corpora of the same size, so

we used approximately 150k words for learning and 9k words for the test. The employed vocabularies are also the same size: they consist of the most frequent 2500 words (including the virtual unknown word noted UNK) constituting each of the learning corpora. Witten-Bell smoothing is applied to all models. The evaluation results in terms of perplexity and entropy are given in Table 8.

Table 8. Perplexity and Entropy Performance Including UNK

n	2-grams		3-grams		4-grams	
	Perplexity	Entropy	Perplexity	Entropy	Perplexity	Entropy
Corp1	279.35	8.13	286.29	8.16	317.40	8.31
Corp2	333.99	8.38	351.27	8.46	387.41	8.60

There is a very substantial difference in terms of perplexity between the two test corpora. The rate of words out of the vocabulary is also very low (27.53%), while that of Corp2 is 40.79%. These results can be explained by the fact that the vocabulary[1] used in the Al-Khabar corpus is much smaller than that of the CAC corpus. Indeed, the Al-Khabar corpus is a collection of texts of local information, while the CAC corpus texts refer to different domains. Therefore, we can state that a language model built on a corpus of weak homogeneity is less efficient than a model calculated on a corpus of high homogeneity. We can then deduce from these results that the nature of the training corpus is a factor that affects the performance of language models.

6. DISTANCE-BASED LANGUAGE MODEL

In any natural language, a word is not only related to its immediate neighbors but also to distant words [18]. For n-gram models, language modeling is reduced to a single relation between the word to predict and its immediate and contiguous history. To illustrate that, let us consider the following sentence:

الوردة التي أهديتني ذابلة

The histories الوردة التي أهديتني and الوردة should be similar in terms of prediction of the word ذابلة. Unfortunately, a classical language model is unable to take account of this phenomenon. Only a distance-based language model may use adequately the word الوردة for the prediction of ذابلة.

Figure 4 and Figure 5 show the kind of relationship taken into account by classical language models and the distance language model, respectively [19].

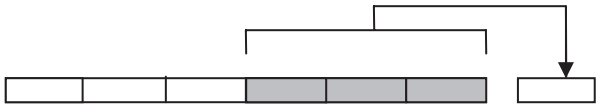

Figure 4: Relationship taken into account by classical language models

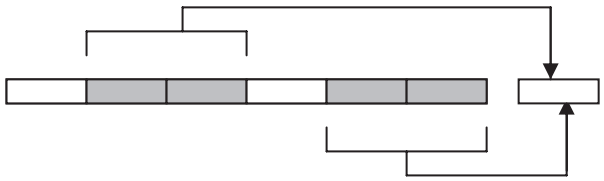

Figure 5: Example of relationships taken into account by distance-based language model

A distant n-gram estimates the probability of a word W_i given a sequence of words $h_d = W_{i-n+1-d} .. W_{i-1-d}$ located exactly at *d* words before W_i. .

In this case, the probability of a word W_i is

$$P_d(W_i/h_d) = \frac{N_d(h_d W_i)}{N(h_d)} \tag{5}$$

where $N_d(h_d W_i)$ is the occurrence of $h_d W_i$. Obviously, due to the distance between h_d and W_i, such a model is less powerful, when it is used alone, than a baseline n-gram. However, distant language models are more efficient when they are combined with classical n-grams [20]. Also, a distant language model holds into account only relations of distance equal to *d*. However, the relationship between the constituents of a sentence can emerge at

[1] By vocabulary we mean different words constituting the corpus.

various distances [21]. That is why it is necessary to combine several n-gram distant models of different distances. We, therefore, computed several language models with different distances (d = 1, 2, 3, 4) on the corpus (Corp2) and using the same vocabulary of the most frequent 2500 words. We then made several linear combinations of classical n-gram and distant n-gram models (for n = 2, 3 and 4). Table 9 illustrates the results of these experiments. Let us remark that a distance $d = 0$ corresponds to a classical n-gram model.

The notation MC(i,j,...) is used to describe the model obtained by combining models corresponding to distances $d = i, d = j, d = ...$

Table 9. Perplexity's Values of MC (i, j...) Models

n	baseline	MC(0,1)	MC(0,1,2)	MC(0,1,2,3)	MC(0,1,2,3,4)
2	333.99	152.18	108.25	89.65	80.08
3	351.27	159.01	116.33	95.85	84.78
4	387.41	172.06	122.04	100.19	88.37

These results show that the use of distant n-gram models greatly improves the perplexity of Arabic n-gram models.

7. COMPARATIVE STUDY OF ARABIC AND FRENCH N-GRAM MODELS

French is used as a second language in several countries in Africa. In the Maghreb, it is an administrative language and commonly used, though not on an official basis in the Maghreb states: Mauritania, Algeria, Morocco, and Tunisia.

In Algeria, French is still the most widely studied foreign language, is widely spoken, and is widely used in media and business. In this section, we investigate a comparative study of Arabic and French n-gram model performances. To our knowledge, this kind of study has never been done and we would like to investigate the differences between these two languages over their respective n-gram models. The main objective of this study is to analyse by using statistical methods the difficulty of Arabic in comparison to French. Indeed, several works on Arabic claim that this language is more difficult than others. We will then investigate which kind of language model is necessary to obtain an equivalent French performance. For our experiments, corpora used for Arabic are extracted from the CAC corpus [17]. For French, the models were trained on corpora extracted from *Le Monde, a* French newspaper. We decided to use identical sizes so that the results could be comparable. Therefore, each training corpus contains 580K words. For the test, each one is made of 33K words and the vocabulary consists of the most frequent 3000 words of the training corpus.

Several Arabic and French n-gram language models are computed in order to study their pertinence for these languages. A few smoothing techniques are tested in order to find out the best model for both of them. The results obtained are listed in Table 10 and Table 11.

Table 10. Performance of Arabic n-Gram Models in Terms of Perplexity and Entropy With Several Smoothing Methods

n	Good-Turing		Witten-Bell		Linear	
	Perplexity	Entropy	Perplexity	Entropy	Perplexity	Entropy
2	326.14	8.35	310.17	8.28	346.68	8.44
3	265.03	8.05	240.41	7.91	292.07	8.19
4	233.97	7.87	204.44	7.68	261.84	8.03

Table 11. Performance of French n-Gram Models in Terms of Perplexity and Entropy With Several Smoothing Methods

n	Good-Turing		Witten-Bell		Linear	
	Perplexity	Entropy	Perplexity	Entropy	Perplexity	Entropy
2	157.40	7.30	154.89	7.28	170.35	7.41
3	141.02	7.14	140.35	7.13	170.26	7.41
4	144.55	7.18	151.12	7.24	182.50	7.51

Let us notice that the perplexity of the French language is lower than the Arabic one, whatever the order of the Arabic n-gram. In other words, the Arabic language seems to be more perplexed. This can be mainly explained by the fact that Arabic texts are rarely diacritized.

Diacritics are short strokes placed above or below the preceding consonant. They indicate short vowels and other pronunciation phenomena, like consonant doubling [22]. The absence of this information leads to many identical looking word forms (*e.g.*, the form ktb **كتب** (write) can correspond to **كَتَبَ** *(wrote),* **كُتُب** *(books),* …) in a large

variety of contexts, which decreases predictability in the language model.

In addition, Arabic has a rich and productive morphology that leads to a large number of probable word forms. This increases the out of vocabulary rate (37.55%) and prevents the robust estimation of language model probabilities.

Also, for French, trigram models are the most appropriate, whatever the smoothing technique used. For Arabic, it seems that n-gram models of higher order could be more efficient. This observation is confirmed by the values given in Table 12. True enough, the 5-gram models are more efficient for Arabic, whatever the smoothing technique. The Witten-Bell discounting method seems especially to be the most powerful for this language. This result is understandable. In fact, when the corpus is small, in general, the suggested smoothing method is Witten-Bell [23]. For French, these results are not confirmed (Table 13).

Table 12. Performance of Arabic Higher Order n-Gram Models in Terms of Perplexity and Entropy

n	Good-Turing		Witten-Bell		Linear	
	Perplexity	Entropy	Perplexity	Entropy	Perplexity	Entropy
5	229.29	7.84	184.95	7.53	258.07	8.01
6	238.75	7.95	176.99	7.47	279.56	8.13
7	254.96	7.99	173.73	7.44	323.50	8.34
8	269.06	8.07	172.47	7.43	415.3	8.70
9	279.07	8.12	172.35	7.43	∞	∞

Table 13. Performance of French Higher Order n-Gram Models in Terms of Perplexity and Entropy

n	Good-Turing		Witten-Bell		Linear	
	Perplexity	Entropy	Perplexity	Entropy	Perplexity	Entropy
5	148.31	7.21	159.48	7.32	191.59	7.58
6	151.02	7.24	164.30	7.36	198.45	7.63
7	152.04	7.25	166.05	7.38	∞	∞
8	152.37	7.25	166.67	7.38		
9	152.65	7.25	166.87	7.38		

In order to summarize these results, we illustrate them with the curve of Figure 6. In general, models smoothed with Good Turing or Witten-Bell are the most appropriate. The linear smoothing technique provides infinite values from $n = 9$ for Arabic and $n = 7$ for French.

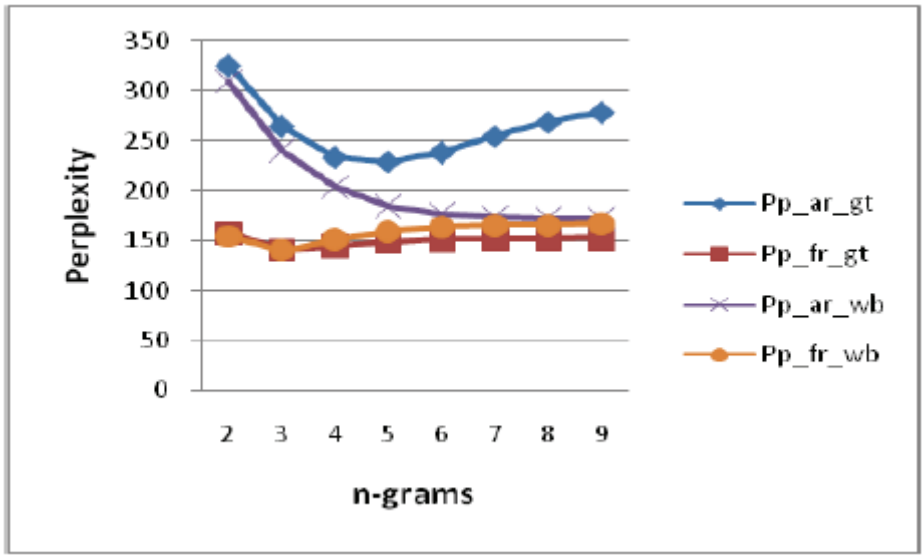

Figure 6: Comparison of perplexities obtained for Arabic (ar) and French (fr) n-gram language models with Good Turing (gt) and Witten Bell (wb) smoothing techniques

We can see that the variation in terms of perplexity is very important from one Arabic model to another. However, the difference is not as wide for French.

The Good Turing technique gives the best perplexity for French (Pp-fr-gt), whereas Witten-Bell is the most efficient for Arabic (Pp-ar-wb). The perplexity is stabilized, only with this smoothing technique, starting from $n = 8$. Note also that with this value of n and only with Witten-Bell smoothing, the models' performances for both

languages are close.

Influence of the vocabulary size

To strengthen these results, we have carried out several experiments by varying the size of the training vocabulary. Figure 7 gives the perplexity values of the most efficient models of Arabic and French.

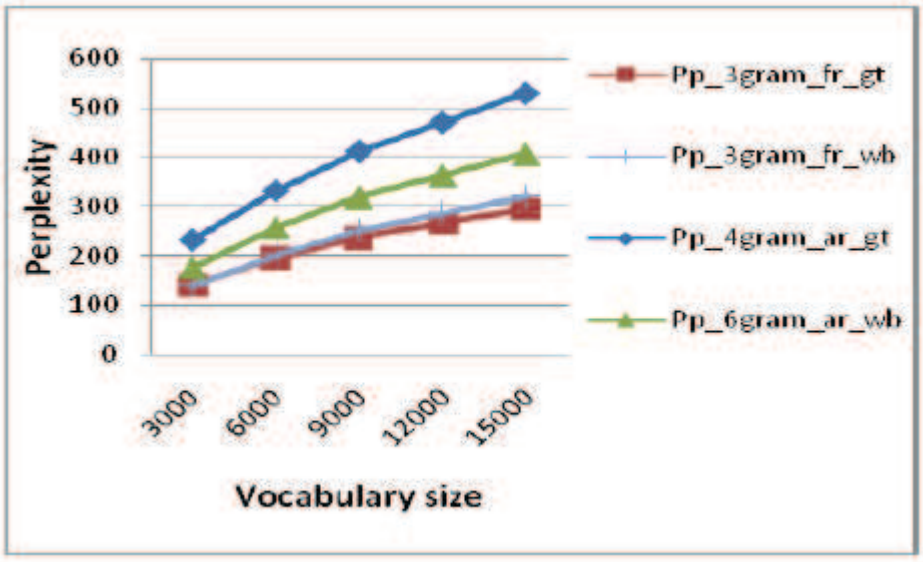

Figure 7: Evolution of perplexity for both languages depending on the vocabulary size

Once again, trigram models with Good Turing smoothing (Pp-3gram-fr-gt) are the most efficient for French whatever the vocabulary size. For Arabic, the n-gram models smoothed with Witten-Bell are the most effective whatever the size of the vocabulary.

It is worth noting also that the change in the size of the vocabulary has a direct influence on the number of words Out Of Vocabulary (OOV) (Figure 8). However, this increase in vocabulary leads to a significant degradation of language models, especially Arabic ones (Figure 7).

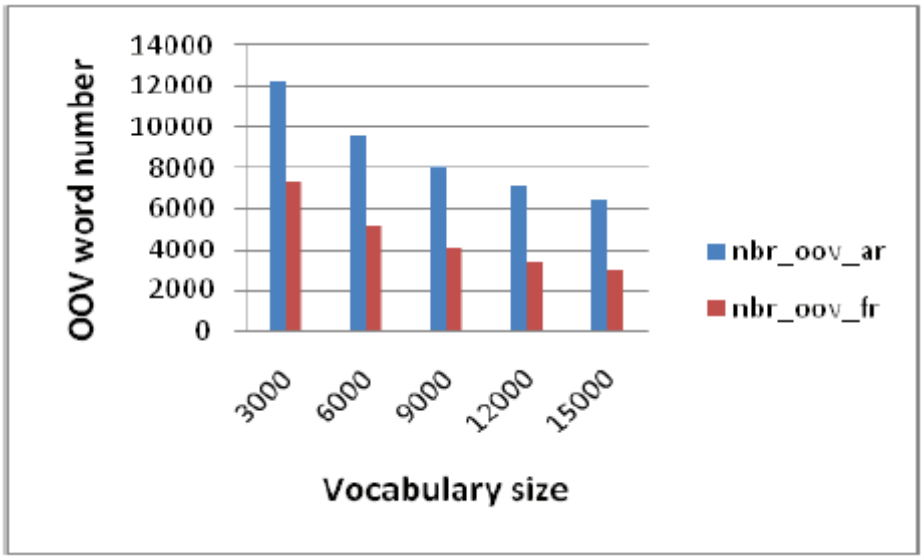

Figure 8: Variation in the number of words OOV for Arabic (nbr-oov-ar) and French (nbr-oov-fr) depending on the size of the training vocabulary

8. COMPARISON WITH OTHER LANGUAGES

The objective of this study is to check if languages belonging to the same language family have an influence on the behavior of language models; especially that they have a remarkable structural relationship. For this purpose, we computed several n-gram models. We considered, from the Indo-European family, three languages: Portuguese,

English and Greek; and Finnish as a language belonging to another family.

The corpora used for this experiment are extracted from the EUROPARL corpus [24]. The parallel Europarl corpus is extracted from the proceedings of the European Parliament. It includes versions in 11 European languages: Romanic (French, Italian, Spanish, Portuguese), Germanic (English, Dutch, German, Danish, Swedish), Greek, and Finnish.

Results of language models

By examining the results of tables (14, 15, 16, and 17), we can observe that the perplexity values of the various models are more-or-less close, except for Finnish for which the difference is remarkable. We can also see that the trigram models are once again most powerful whatever the smoothing technique used. Especially, the models smoothed with the Good-Turing method are most powerful.

Table 14. Portuguese n-Gram Models Performances

n	Good-Turing		Witten-Bell		Linear	
	Perplexity	Entropy	Perplexity	Entropy	Perplexity	Entropy
2	120.53	6.91	120.16	6.91	129.14	7.01
3	101.79	6.67	102.96	6.69	122.69	6.94
4	105.61	6.72	110.74	6.79	134.51	7.07
5	109.88	6.78	117.04	6.87	144.35	7.17
6	112.43	6.81	119.59	6.90	153.08	7.26
7	113.67	6.83	120.22	6.91	∞	∞
8	114.31	6.84	120.42	6.91		
9	114.64	6.84	120.49	6.91		

Table 15. English n-Gram Models Performances

n	Good-Turing		Witten-Bell		Linear	
	Perplexity	Entropy	Perplexity	Entropy	Perplexity	Entropy
2	116.02	6.89	116.36	6.86	124.21	6.81
3	101.50	6.67	103.02	6.69	120.94	6.77
4	104.21	6.70	109.05	6.77	131.94	6.90
5	108.30	6.76	114.32	6.84	142.27	7.02
6	111.05	6.80	116.45	6.86	151.02	7.12
7	112.15	6.81	117.10	6.87	∞	∞
8	112.68	6.82	117.23	6.87		
9	112.91	6.82	117.26	6.87		

Table 16. Greek n-Gram Models Performances

n	Good-Turing		Witten-Bell		Linear	
	Perplexity	Entropy	Perplexity	Entropy	Perplexity	Entropy
2	99.99	6.64	99.95	6.64	107.78	6.75
3	86.04	6.43	85.85	6.42	103.06	6.69
4	88.48	6.47	90.38	6.50	111.63	6.80
5	92.13	6.53	95.73	6.58	120.22	6.91
6	94.86	5.57	98.67	6.62	127.53	6.99
7	95.97	6.58	99.73	6.64	∞	∞
8	96.36	6.59	100.00	6.64		
9	96.60	6.59	100.08	6.64		

Table 17. Finnish n-Gram Models Performances

n	Good-Turing		Witten-Bell		Linear	
	Perplexity	Entropy	Perplexity	Entropy	Perplexity	Entropy
2	269.59	8.07	262.30	8.04	297.41	8.22
3	251.25	7.97	251.60	7.97	306.91	8.26
4	255.12	8.00	270.33	8.08	329.60	8.36
5	261.21	8.03	285.62	8.16	350.76	8.45
6	266.15	8.06	294.10	8.20	369.43	8.53
7	269.14	8.07	298.77	8.22	409.29	8.68
8	271.04	8.08	300.50	8.23	∞	∞
9	271.92	8.09	301.22	8.23		

We summarize in Figure 9 the performances of statistical models of all languages.

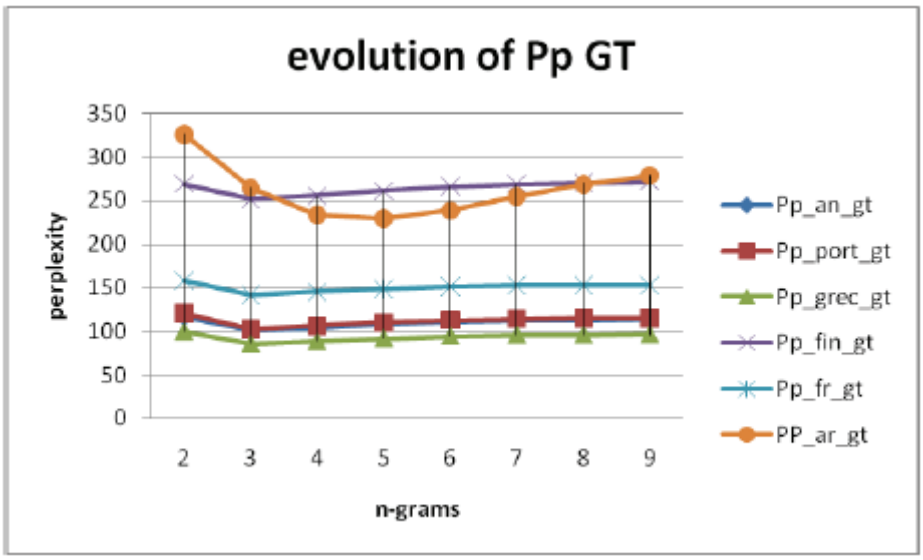

Figure 9: Comparison of the perplexity values of the different language models smoothed with Good-Turing technique (gt)

Indeed, the curves relating to French, Portuguese, English, and Greek are very similar: they all start with a perplexity more-or-less high for $n = 2$, reach a minimal value for $n = 3$, then from $n = 4$ they start increasing. The perplexity values of Finnish models are higher compared to perplexities obtained for Indo-European languages. However, starting from $n = 8$, they coincide with those of Arabic models smoothed with Good-Turing. These values can be explained by the agglutinate nature of the Finnish language. This characteristic increases the number of words out of vocabulary and consequently prevents the robust evaluation of the probabilities of the language model. Figure 10 illustrates the variation in number of out of vocabulary words for all considered languages.

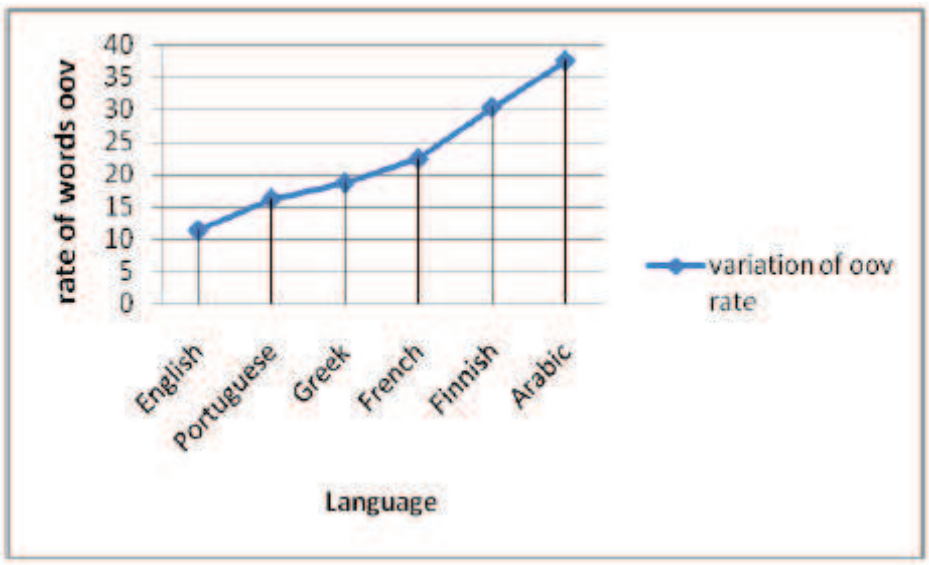

Figure 10: Variations in rates of words out of vocabulary

9. CONCLUSION

In this study, we proposed to investigate statistical language models for Arabic. First, several experiments using different smoothing techniques have been carried out on a small corpus extracted from a daily newspaper. The sparseness of the data leads us to investigate other solutions without increasing the size of the corpus. A word segmentation technique has been used in order to increase the statistical viability of the corpus. This leads to a better performance in terms of normalized perplexity. We think that even with a large corpus, segmentation is necessary. In fact, a lot of words in Arabic are constructed from patterns which are used as generative rules. Each pattern indicates not only how to build a word but gives the syntactic role of the generated word.

The second experiments were conducted to demonstrate the influence of the nature of the training corpus on the performance of language model. We also showed that the use of distant n-gram models is also effective for the Arabic language. In future work, we hope to combine morpheme models and distant n-grams to assess their contribution.

Finally, we investigated a comparative study of Arabic n-gram language models with several other languages. For these languages, trigram models are most appropriate whatever the smoothing technique used. For Arabic, the n-gram models of higher order smoothed with the Witten-Bell method are more efficient.

As in other morphologically rich languages, the large number of possible word forms entails problems for robust language model estimation. It is, therefore, preferable, for Arabic, to use morpheme-like units instead of whole word forms as language modeling units.

As a continuation of this work, we plan to study Arabic language models built on a fully diacritized corpus. Indeed, it would be interesting to compare their performances with those of models computed on a corpus without diacritics. The problem is that it is difficult to find a fully diacritized corpus of sufficient size. A solution would be to use automatic diacritization tools.

REFERENCES

[1] W. Kim and S. Khudanpur, "Cross-Lingual Lexical Triggers in Statistical Modeling", in *Proceeding of the Conference on Empirical Methods in Natural Language Processing*, *New York*, 2003, pp. 1–5.

[2] K. Meftouh, K. Smaili, and M. T. Laskri, "Arabic Statistical Modeling", in *Proceedings of 9e Journées Internationales d'Analyse Statistique des Données Textuelles*, *Lyon, France*, 2008, pp. 837–844.

[3] K. Meftouh, K. Smaili, and M. T. Laskri, "Comparative Study of Arabic and French Statistical Language Models", in *Proceedings of the International Conference on Agents and Artificial Intelligence ICAART, Porto, Portugal*, Jan 2009.

[4] K. Hayder Al Ameed and O. Shaikha Al Ketbi *et al.*, "Arabic Light Stemmer: New Enhanced Approach", in *Proceeding of the Second International Conference on Innovations in Information Technology IIT'05*, 2005.