MorphTagger: HMM-Based Arabic Segmentation for Statistical Machine Translation

Saab Mansour

Human Language Technology and Pattern Recognition
Computer Science Department
RWTH Aachen University
Aachen, Germany
mansour@cs.rwth-aachen.de

Abstract

In this paper, we investigate different methodologies of Arabic segmentation for statistical machine translation by comparing a rule-based segmenter to different statistically-based segmenters. We also present a new method for segmentation that serves the need for a real-time translation system without impairing the translation accuracy.

1. Introduction

Data-driven methods have been applied very successfully within the Machine Translation (MT) domain since the early 90s. Significant improvements in the field have been made through advances in modeling, availability of larger corpora and more powerful computers. The requirement for acceptable translation results has led to the development of systems trained on millions of sentence pairs. Nevertheless, often, a requirement for these systems is the capability to process text in "real-time", i.e. without complex preprocessing and translation setup that would need minutes or even hours for a single document.

One of the major problems of statistical models is the data sparseness problem which consequently forces researchers to develop statistical models which are trained on local or limited context. In order to lessen the data sparseness problem for the task of Arabic Statistical MT (SMT), we apply the well studied method of segmentation as a preprocessing step. A word in Arabic may be composed of prefixes, a stem and suffixes which are expressed as standalone words in many languages. Those attachment clitics include prepositions and subjective, objective and possessive pronouns. Except reducing the data sparseness problem, segmentation results in minimizing the differences between Arabic and the target language, smaller vocabulary size and less out-of-vocabulary (OOV) words. An example of Arabic segmentation is shown in Figure 1 where the Arabic words are depicted with the corresponding Buckwalter transliteration[1]. One observation from this figure is that using segmentation, a better one-to-one correspondence between English and Arabic is achieved. In this work, we compare the performance of several segmenters on several SMT tasks. We also introduce a new segmentation method that answers the needs of a real-time translation system without impairing the translation quality.

This paper is organized as follows. Related work on Arabic segmentation is presented in Section 2. In Section 3, we discuss the problems of Arabic SMT, present the solution of segmentation and existing tools to perform this task. In Section 4, we present the MorphTagger architecture, modelling and implementation details and speed comparison to existing segmentation tools. The different settings will be evaluated in Section 5, where we show experiments on various tasks having Arabic as the source language. A discussion of the results and further examples including final remarks and future work are given in Section 6.

2. Related work

Arabic segmentation for the task of SMT was already successfully applied in previous work. [1] uses a language model to select among possible segmentations for translating Arabic into English. They report improvements for small tasks, but no improvements for big tasks. [2] apply the MADA tool for Arabic-English machine translation. MADA selects among Buckwalter Arabic Morphological Analyzer (BAMA) analyses using a combination of Support Vector Machine (SVM) classifiers. Their work is mainly focused on comparing different segmentation schemes. [3] develop a Finite State Transducer (FST) based segmenter and apply it to Arabic-English SMT and later on to Arabic-French SMT (cf. [4]). Their work also compares to an SVM based segmenter presented by [5] and shows superior results for small tasks but inferior ones for large tasks. [6] apply a Conditional Random Fields (CRF) segmentation method for Arabic to English translation. They show that a reduced morpheme segmentation, where they apply a statistically trained model to delete morphemes, outperforms a full morpheme segmentation.

[1] http://www.qamus.org/transliteration.htm

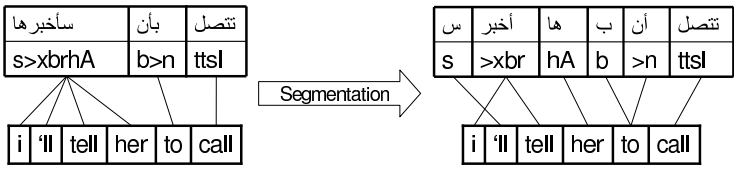

Figure 1: Arabic segmentation example: Arabic words are accompanied by the Buckwalter transliteration and the alignment to the words on the English side

In this work, we perform consistent comparison of several segmentation methods on several translation tasks. We also present a new segmentation method that is quick enough to be used in a real-time translation system without impairing the accuracy. Furthermore, the new method shows consistent improvement on both small and large scale translation tasks.

3. Arabic segmentation

Written Modern Standard Arabic (henceforth *Arabic*) is known for its complex morphology and ambiguous writing system. For the task of SMT, Arabic holds the following properties:

- high rate of inflection causing high percentage of Out-Of-Vocabulary (OOV) words. In addition to the inflection expressing different grammatical categories found in English (gender, number, ...), Arabic inflection includes the generation of words using the root-pattern constructor and the attachment of clitics (to a stem) which appear as stand-alone words in many other languages. An example is given in Figure 3. The first sentence in this figure is a hypothesis generated by our translation system without Arabic segmentation. The second hypothesis is generated by a system which includes Arabic segmentation, causing one OOV word to be resolved.

- high ambiguity due to the lack of vowels in written Arabic. The increase of ambiguity is expressed in the increased number of possible translations per word, but, in addition, it is expressed in the possible segmentations of the word which eventually affects the corresponding translations. An example is given in Figure 3.

- one word in Arabic often corresponds to more than one word in traditional target languages such as English and French, posing a problem to the alignment models. An example is given in Figure 1. In this figure, we can see that some Arabic words could be aligned to more than one word in English. This causes a problem to the traditional alignment models which are found in the basis of most of the state-of-the-art SMT systems.

A well studied solution to the problems mentioned above is Arabic word segmentation. Splitting an Arabic word into its corresponding prefixes, stem and suffixes lessens the number of OOV words, resolves some of the ambiguous Arabic words and generates more one-to-one correspondences between the Arabic side and the target language side which can be easily captured by the IBM alignment models.

- non-segmented (HYP1) vs segmented (HYP2) Arabic hypotheses:
  - HYP1: sorry , i have to UNKNOWN_نرفض UNKNOWN_عرضك .
  - HYP2: sorry , i have to UNKNOWN_نرفض your offer .
- different segmentations of the word للجنة *lljnp* due to the lack of vocalization:
  - لِلَجنَة *li+lajnap* 'to a committee'
  - لِلَّجنَة *li+l+lajnap* 'to the committee'
  - لِلجَنَّة *li+l+jn˜ap* 'to the heaven'

Figure 3: Arabic difficulties for SMT: Examples

As mentioned in Section 2, some work has been done on Arabic segmentation for SMT. The FST tool presented by [3] inherently suffers from ambiguous words which are not segmented in the approach. A problem of the FST method is that it achieves improved results over a statistical segmenter for a small task, but inferior results for a large task. Another well known segmentation tool for Arabic is the MADA tool. [2] perform a comparison between the different segmentation schemes supported by MADA, but a comparison to other techniques is not included. Another problem of the MADA tool is the slow speed of the segmentation process. MADA applies several SVM classifiers to classify different grammatical categories of the words and then combines those classifications to infer full morphological disambiguation. (non-linear) SVM classification has the time complexity of $\theta(n \cdot |SV|)$, where n is the number of words in the text being segmented and $|SV|$ is the number of support vectors generated in the training phase. $|SV|$ is upper bounded by the number of training examples. In the case of MADA this is in the magnitude of 10^5 as it is trained on the Arabic Treebank.

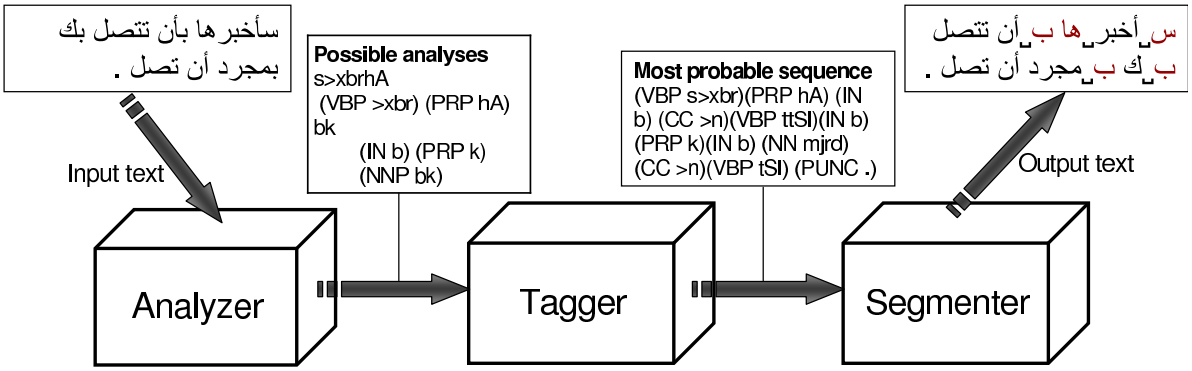

Figure 2: MorphTagger segmenter architecture

In this work, we present a Hidden-Markov-Model (HMM) segmenter for Arabic. The motivation behind the development of this tool is the need for a segmenter which achieves comparable accuracy to MADA, but retains a speed level similar to the FST segmenter and which is acceptable for real-time translation systems.

4. MorphTagger: HMM-based segmenter

MorphTagger is a general architecture for Part-Of-Speech (POS) tagging of natural languages. The architecture was first proposed in [7] where it was applied for the task of POS tagging of Hebrew. [8] adapted the architecture to the Arabic language. In this work, MorphTagger was adapted to the SMT task, adding a segmenter level and few normalization rules that are appropriate for translation, in addition to speed enhancements of the software. The architecture is similar to [2] where one selects a specific analysis from the output of a morphological analyzer. The architecture is visualized in Figure 2.

First, the Arabic input sentence goes through a morphological analyzer, which outputs for each word all possible analyses. Each analysis includes a sequence of pairs of a segment and the corresponding POS tag.

The Tagger component outputs the most probable tagging sequence according to the model. Then, we infer the corresponding segments by matching the tagging sequence to the analyses. Since the process of matching the corresponding segments is ambiguous, we simply use the heuristic of choosing the most probable morpheme given the tag. We believe that the heuristic is sufficient for the problem at hand as the ambiguity mainly occurs for variations of Arabic letters such as Alef (أ, إ, آ, ا), and is rarely observed for segmentation boundaries. The variation of the letters could be modeled by a finer grained tag set, but this modeling is not used in this work.

The Segmenter component is then responsible for the choice of which morphemes should be split. This component is realized by rules which are selected manually. The segmenter also applies several normalization steps which proved to be helpful for SMT.

4.1. Modeling framework

To model the Tagger component in MorphTagger, we use a standard HMM disambiguation, while limiting the choice of possible analyses to the set provided by the morphological analyzer. We denote our set of observed word sequence by $w_1^N = w_1, ...w_n, ...w_N$, $a(w_n)$ is the set of analyses for the word w_n provided by the morphological analyzer and $a(w_1^N)$ is the set of the whole word sequence analyses.

The problem at hand is to find the most probable POS tags $t_1^N = t_1, ..., t_n, ...t_N$ associated with w_1^N:

$$t_1^N = \underset{\tilde{t}_1^N \in a(w_1^N)}{\operatorname{argmax}} Pr(\tilde{t}_1^N | w_1^N) \quad (1)$$

Using the Bayes decision rule and the bigram HMM model assumptions, we can rewrite 1 as:

$$t_1^N = \underset{\tilde{t}_1^N \in a(w_1^N)}{\operatorname{argmax}} \left\{ \prod_{n=1}^{N} [p(w_n|\tilde{t}_n) \cdot p(\tilde{t}_n|\tilde{t}_{n-1})] \right\} \quad (2)$$

The language model parameters $\{p(t_n|t_{n-1})\}$ and the lexical model parameters $\{p(w_n|t_n)\}$ are estimated on the segment level using Maximum Likelihood Estimation (MLE) followed by an array of smoothing techniques explained in [8]. As we are working on the segment level, the lattice that the HMM model is traversing might have paths with different lengths. The MLE estimates seem to work quite well in this case, as segmented words are formed from a stem and clitics, where the clitics have a high (lexical) probability. Thus, the probability of the stem will be the major factor in the probability of a segmented word.

4.2. Implementation details

To implement MorphTagger for Arabic, we use the Buckwalter Arabic Morphological Analyzer v1.0[2], a rule based

[2]LDC Catalog No. LDC2002L49

Table 1: Segmentation speed measured in words per second

	speed [w/s]
FST	4 500
MADA	70
MorphTagger	1 500

analyzer, with 80 000 lexicon entries. The POS model is a standard Markov Model Tagger trained over the Arabic Treebank Part 1 v3.0[3] (150 000 tokens). We estimate the probabilities of the model for segments and not words, because it achieves better POS tagging and segmentation accuracies as reported in [7]. The disambiguator is implemented by wrapping around the SRILM[4] toolkit. The Segmenter component splits prepositions (excluding the Arabic determiner) and possessive and objective pronouns (this is the so-called *ATB* scheme originally used in the Arabic TreeBank). As mentioned before, the segmenter also performs few normalization steps, most noticeable undoing some rewriting rules when attachment is involved. Reverted characters include: *(i)* 'alif maksura: reverted to the original form when a preposition ending with 'alif maksura is split from a suffix (yX→Y+X); *(ii)* feminine marker: reverted to its original form when a noun is split from a suffix (tX→p+X); and *(iii)* Arabic determiner: is unhidden when preceded by ل *l* 'to' preposition (llX→l+Al+X).

Due to the way MorphTagger is implemented, we achieve the following three desirable advantages:

- state-of-the-art segmentation accuracy [5]
- training and tagging are fast (linear in corpus size)
- appropriate for real-time systems

4.3. Segmentation speed results

In Table 1 we present a comparison between the speed of the different segmenters. The speed is measured in units of words per second ([w/s]). From this table we see that the MADA tool can not be applied in a real-time manner. For example, our real-time Arabic-French SMT system (will be presented in Section 5) is running at the speed of 100 [w/s], making the MADA segmenter slower than the translation system and non-appropriate for such applications.

5. Translation experiments

In this section, we evaluate the translation performance of the MorphTagger segmenter. We compare the results of MorphTagger to the MADA and the FST segmenters. The baseline system was built using a state-of-the art phrase-based MT system described in [9]. We use the standard set of models with phrase translation probabilities for source-to-target and target-to-source direction, smoothing with lexical weights, a word and phrase penalty, distance-based reordering and an n-gram target language model.

Two evaluation tasks were used to experiment with the performance of MorphTagger: the BTEC 2008 Arabic-English task and the QUAERO 2009 Arabic-French task[6]. Corpus statistics of the BTEC and the QUAERO tasks are given in Table 2 and Table 3 respectively. The tables include statistics of the training corpora and test sets used, calculated over the various segmentation methods. We also include statistics of a simple tokenizer (TOK) for Arabic which splits on punctuations, to serve for comparison purposes to the other segmenters. For the QUAERO task, the development and test sets consist of one reference on the French side, the CESTA_RUN2[7] test has four references. The test sets of the BTEC task consist of 16 references. We can already see from the number of running words in those tables that the segmented Arabic text is more similar to English. We also see a notable reduction in OOV words of about 40 percent in the BTEC task and up to 75 percent in the QUAERO task. One interesting point to notice about the OOV figures is that the FST method is sometimes performing worse than a simple tokenizer. The reason behind this is that the FST method restricts stems to those seen in the corpus, therefore preventing segmenting words that include unseen stems. This causes inconsistencies in the segmentations between the train and the test sets.

The results of the QUAERO 2009 task are summarized in Table 4. The results are truecased (case). Real-time systems use a monotone decoder and a smaller language model (4-gram instead of 6-gram in the offline systems). Offline systems include reordering and bigger language model. In terms of speed, real-time systems translate more than 100 words per second, whereas the offline systems are running at less than one word per second.

In the real-time systems results, we see that MorphTagger, in comparison to MADA, achieves modest improvements of +0.3% BLEU and comparable TER on both Test and CESTA_RUN2 test sets. The FST method is performing much worse on CESTA_RUN2, probably due to the OOV problem mentioned earlier.

For the offline systems, we added a TOK system were Arabic input was only tokenized. As in previous work, we see that Arabic words segmentation helps over the TOK only method, with improvements up to +1.2% BLEU and -1.5% TER on the Test set. When comparing the three segmenters, the BLEU tendency on the test sets is quite similar to the real-time systems results. From the other hand, MorphTagger achieves significantly better TER results. We hypothesize that this might be due to the different normalization done in the segmenters, seemingly resulting in better lexicon models

[3] LDC Catalog No. LDC2005T02

[4] http://www-speech.sri.com/projects/srilm/

[5] see [8] for details

[6] The QUAERO project website: http://www.quaero.org. Note that the data is available for the project partners only.

[7] CESTA_RUN2 is the official test set of the second CESTA evaluation campaign held in October 2005.

Table 2: AR-EN BTEC 2008: Corpus statistics

		Arabic				English
		TOK	FST	MADA	MorphTagger	
Train	Sentences	24K				
	Running Words	158K	184K	186K	187K	240K
	Vocabulary	19K	14K	14K	14K	8K
IWST04 (dev)	Sentences	500				
	Running Words	2 659	2 933	3 149	3 152	-
	OOV	142	190	82	91	-
IWST05	Sentences	506				
	Running Words	2 566	2 994	3 041	3 063	-
	OOV	149	96	91	96	-
IWST08	Sentences	507				
	Running Words	2 585	2 994	3 075	3 064	-
	OOV	182	125	111	114	-

Table 3: AR-FR QUAERO 2009: Corpus statistics

		Arabic				French
		TOK	FST	MADA	MorphTagger	
Train	Sentences	7.6M				
	Running Words	150M	170M	175M	178M	196M
	Vocabulary	638K	380K	422K	380K	300K
Dev	Sentences	2121				
	Running Words	50 389	57 264	58 335	58 516	-
	OOVs (run.)	337	289	176	185	-
Test	Sentences	2202				
	Running Words	49 617	56 065	57 235	57 535	-
	OOVs (run.)	318	296	180	191	-
CESTA_RUN2	Sentences	824				
	Running Words	19 329	22 019	22 524	22 895	-
	OOVs (run.)	118	224	44	56	-

and lexical choice for the MorphTagger segmenter. Looking at the translations, we see that few differences are the result of different segmentations, especially between MADA and MorphTagger as they use the same segmentation scheme. A more significant difference between the segmenters might be due to the different normalization they apply. In MADA, in addition to the normalizations mentioned in Section 4, many irregular word writings are collapsed to one form.

Translation examples are given in Table 6. In the first sentence, the FST does not split the Arabic preposition ب *b* 'in', and MADA splits the feminine marker ة *p* wrongly. The translations of MorphTagger and MADA are similar, indicating that MADA could recover from its segmentation error, whereas the FST is suffering from one unknown word كيث *kyv* 'Keith' because it wrongly segmented it in the training data. In the second example, MADA does not segment the word وفى *wfY* 'and in', which then can also mean 'Acquitté', causing a wrong translation.

The BTEC task results are summarized in Table 5. The results ignore casing information but include punctionations (nocase+punc). From this table, we see a similar tendency of improvement as was observed in the QUAERO task results. The three segmenters are improving on the test sets over the simple tokenizer. Whereas, both MADA and MorphTagger are performing better than the FST method, especially on IWSLT05 test set, where improvements of around +0.8% BLEU and -0.4% TER were observed. MorphTagger has a slight edge over MADA on the IWSLT08 set, where it had improvement of +0.5% in BLEU and -0.5% in TER.

6. Conclusions and summary

In this work, we compared and evaluated Arabic segmenters for the task of Arabic statistical machine translation. We started out by comparing two available segmenters, an FST rule-based segmenter and the MADA tool — an SVM-based statistical classifier. The FST segmenter suffers from inferior translation results over large tasks when compared to a statistical segmenter and MADA performs too slow to be incorporated into a real-time SMT system. To combine the best of both worlds, we adapt a Hidden-Markov-Model Part-Of-Speech tagger to the segmentation task and plug it into the translation system as a preprocessing step. Being an HMM disambiguator, the POS tagging time complexity is linear in corpus size and proves to be comparable to the speed of the FST method and applicable to real-time systems. Furthermore, the HMM model incorporates context knowledge to infer the output classes, thus resulting in a better, more consistent segmentation result than the FST method.

We compared MorphTagger to the FST and the MADA segmenters and showed improved results on different translation conditions and different test sets.

7. Acknowledgments

This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation and partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-08-C-0110.

8. References

[1] Y.-S. Lee, "Morphological Analysis for Statistical Machine Translation," in *Proceedings of HLT-NAACL 2004: Short Papers*. Morristown, NJ, USA, 2004, pp. 57–60.

[2] F. Sadat and N. Habash, "Combination of Preprocessing Schemes for Statistical MT," in *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, July 2006, pp. 1–8.

[3] A. El Isbihani, S. Khadivi, O. Bender, and H. Ney, "Morpho-Syntactic Arabic Preprocessing for Arabic to English Statistical Machine Translation," in *Proceedings of the Workshop on Statistical Machine Translation*. New York City, June 2006, pp. 15–22.

[4] S. Hasan, A. El Isbihani, and H. Ney, "Creating a Large-Scale Arabic to French Statistical Machine Translation System," in *International Conference on Language Resources and Evaluation*, Genoa, Italy, May 2006, pp. 855–858.

[5] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks," in *HLT-NAACL 2004: Short Papers*. Boston, Massachusetts, USA, May 2 - May 7 2004, pp. 149–152.

[6] T. Nguyen and S. Vogel, "Context-Based Arabic Morphological Analysis for Machine Translation," in *Proceedings of the Twelfth Conference on Computational Natural Language Learning*. Morristown, NJ, USA, 2008, pp. 135–142.

[7] R. Bar-haim, K. Sima'an, and Y. Winter, "Part-of-Speech Tagging of Modern Hebrew Text," *Nat. Lang. Eng.*, vol. 14, no. 2, pp. 223–251, 2008.

[8] S. Manour, K. Sima'an, and Y. Winter, "Smoothing a Lexicon-Based POS Tagger for Arabic and Hebrew," in *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 97–103.

[9] R. Zens and H. Ney, "Improvements in Dynamic Programming Beam Search for Phrase-based Statistical Machine Translation," in *International Workshop on Spoken Language Translation*, Honolulu, Hawaii, Oct. 2008, pp. 195–205.

Table 4: AR-FR QUAERO 2009: Translation results (case)

Real-time systems						
	Dev		Test		CESTA_RUN2	
System	BLEU	TER	BLEU	TER	BLEU	TER
FST	15.5	74.9	15.4	74.8	45.7	53.4
MADA	15.5	73.9	15.5	74.8	47.7	53.0
MorphTagger	15.9	73.9	15.8	74.7	48.0	53.2
Offline systems						
TOK	15.7	74.6	15.3	75.3	45.3	53.6
FST	16.6	73.2	16.3	74.2	47.6	52.1
MADA	16.1	73.7	16.1	74.9	47.8	51.7
MorphTagger	17.1	72.5	16.6	73.5	48.8	49.8

Table 5: AR-EN BTEC 2008: Translation results (nocase+punc)

	IWSLT04 (dev)		IWSLT05		IWSLT08	
System	BLEU	TER	BLEU	TER	BLEU	TER
TOK	55.6	32.8	55.6	32.4	51.8	35.0
FST	52.3	35.2	55.9	32.0	51.7	34.9
MADA	56.0	32.4	56.7	31.6	51.9	35.2
MorphTagger	55.8	32.7	56.8	31.5	52.4	34.7

Table 6: Examples of better translations due to improved Arabic segmentation

Source	كما أود أن أرحب بالدكتورة كيث كارتر
Reference	Je voudrais aussi souhaiter la bienvenue au Dr Keith Carter
FST	كما اود ان ارحب بالدكتور ـة كيث كارتر
	Je souhaite la bienvenue au Dr UNKNOWN_كيث Carter
MADA	كما أود أن أرحب بـ الدكتور ـة كيث كارتر
	Je souhaite la bienvenue au Dr Keith Carter
MorphTagger	كما أود أن أرحب بـ الدكتورة كيث كارتر
	Je souhaite la bienvenue au Dr Keith Carter
Source	وفى كانون الأول / ديسمبر ٢٠٠١ وافق مجلس الوزراء
Reference	En décembre 2001 , le Conseil des ministres a approuvé
FST	وفي décembre 2001 وافق مجلس الوزراء
	En décembre , le conseil des ministres a approuvé
MADA	وفى كانون الأول / ديسمبر ٢٠٠١ وافق مجلس الوزراء
	Acquitté en décembre 2001 , le Conseil des ministres an approuvé
MorphTagger	و_في كانون الأول / ديسمبر ٢٠٠١ وافق مجلس الوزراء
	En décembre 2001 , le Conseil des ministres an approuvé