Persian part of speech tagger based on Hidden Markov Model

Ali Azimizadeh[1], Mohammad Mehdi Arab[2], Saeid Rahati Quchani[3]

Islamic Azad University of Mashhad, Iran
[1]aazimizadeh@yahoo.com; [2]mehdi.arab@gmail.com; [3]Rahati@mshdiau.ac.ir

Abstract

This paper introduces the Persian Part of Speech (POS) tagger, based on the Hidden Markov Models (HMM). This POS tagger is part of the Persian Text-to-Speech (TTS) system called ParsGooyan. The tagger supports some properties of TTS systems, such as Break Phrase Detection, Homograph words Disambiguation, and Lexical Stress Search. A POS lexicon with 61,521 entries and 64,003 trigrams is used as the language model. It is implemented in Festival software and makes use of the Viterbi Decoder provided by Edinburgh Speech Tools. The average overall accuracy for this tagger is 95.11%. The accuracy of the known and unknown words is 96.136% and 60.25%, respectively.

Keywords: POS tagger, HMM, trigram, lexicon, NLP, text normalization, Viterbi decoder.

1. Introduction

Part-of-Speech (POS) Tagging is one of the essential parts of Natural Language Processing (NLP) applications such as text-to-speech (TTS) systems and translation machines. The POS tags contain a significant amount of grammatical information such as quantity, person, and gender about the words and their neighbors (Jurafsky and Martin, 1999).

In TTS systems POS tagging is used for a variety of purposes, such as morphological analysis (Azimizadeh and Arab, 2007), homograph disambiguation, and, in the case of the Persian language, is used to find Ezafe (In Persian ezafe is an unstressed vowel -e or -ye after vowels which is used to link two words in some conditions, ezafe is not written in the orthographic form of the text). In Prosody Synthesis the POS tags are used as feature in some of the fields, for example Break Phrase Detection (Black and Taylor, 1997), Duration and Intonation Model, and also Pitch Contour Estimation (Black et al., 2002). It is also used to obtain annotated corpora; combining automatic tagging with human supervision. These corpora may be used for linguistic research, to build better taggers, or used as statistical evidence for other language-processing related goals (Padr'o, 2004).

POS tagging is the process of choosing the correct grammatical tag for a word based on the context or morphological properties. The Input data in these systems is the input text and the output of them is words accompanied by their POS tags. Taggers are generally divided into three classes: rule-based, statistical-based, and transformation.

Rule-based taggers include a large database of grammatical rules. In this method, the tagger makes a hypothesis and based on its database rules chooses the best case. Statistical taggers are first trained by a labeled corpus. A model is formed from this training, which given a

word outputs the label with the highest probability. The last group is transformation based. Such methods take a hybrid approach based on a variation of the two prior approaches.

In the pioneer languages like English, activities in this field started in the 80's and continue to increase in system accuracy (DeRose, 1988). In Persian there has only been activity in this area in the last few years. One of the first Persian POS taggers is in Assi's work that is in turn based on the Schuetze hypothesis. This hypothesis states that syntactic behavior is reflected in co-occurrence patterns. They hypothesize that, for a given window size, by storing both the left and the right context vector of each word, clustering all similar vectors and then manually annotating each cluster, the POS tags can be estimated by observing the cluster to which the new words belong. (Assi and Abdolhossini, 2000) This system uses a tag-set with 45 tags and performs at 57.7% accuracy.

Another work in Persian is the Orumchian tagger that is based on TnT POS tagger. (Brants, 2000) The TnT tagger follows the Hidden Markov Models (HMM) theory. This tagger has 2.5 million tagged words as training data and the size of the tag-set is 38. It has an overall accuracy is 96.64%. It should be noted that the training data and method used in the Orumchian tagger is also used in the word presented here (Oroumchian et al., 2007).

One of the most recent activities concerns Jabbari and Allison. They use an implementation of Error-Driven Transformation Based Learning. The system learns tagging rules for very coarse part-of-speech categories and subsequently for a full, complex tag-set. (Jabbari and Allison, 2007) This method is formerly used by Brill (Brill, 1995) and Hepple (Hepple, 2000), in English, is a transformation based approach. The structure of this tagger includes a trained learner machine that contains the amount of estimated rules. The training data in Jabbari's work is 1 million tagged words and their tag-set contains 150 tags; the overall accuracy 93%.

The, HMM based, POS tagger presented in this paper is designed with two goals in mind: to increase system accuracy and to provide robustness in various contexts. To achieve these goals the system should be extendable and can be composed with other methods.

This tagger is a part of Persian TTS system called **ParsGooyan** that is implemented in Festival TTS software. It is implemented in this environment by Scheme (SIOD) (Black et al., 2002) script language and by Edinburgh Speech Tools (Black et al., 1998). First Persian text normalization is presented. The implementation of the system is studied in section 3 and optimizations are presented in section 4. Section 5 outlines the results of the system evaluation. The performance of this system is compared against others in the conclusion.

2. Text normalization

In Persian, affixes can be written in three forms: connected, separate, and with half-interspaces. For example, *"می روم" (miravam)* which means "I am going" and *"کتابها" (ketabha)* which means "books" can be written in these three forms:

Connected	Separate	With half-space
میروم	می روم	می‌روم
کتابها	کتاب ها	کتاب‌ها

In Persian orthography, the half space isn't usual. This can create a number of problems in Token-to-Word transformation. Generally, tokenizer systems recognize words by their

interspaces. Therefore affixes, which are not written in the connected form, are counted as two words; this error propagates through the next stages of NLP. To solve this problem the affixes should be attached to their stems everywhere. But this is simply not useful for all of affixes because some affixes are homographs with some words. For example, *"می"* (mi) is an affixes for verbs but can be pronounced (mey) and be meant as "wine". If the system attaches *"می"* where it is being used in its meaning of wine, it can create an error. To differentiate between these cases the context of the words should be taken into consideration.

A two-step tokenizer is designed in this project. First, the Persian letters are transformed to English letters that are then extracted by the tokenizer by space and punctuation characters. In the next step the affixes are reattached to their stems. There are two types of affixes that should be reattached:

1. Those are not similar to stems, e.g.*"تان" "tan"* (yours).
2. Affixes that are homograph with stems, e.g. *"تر"* can be *"tar"* (suffix) and *"tar"* (wetness).

A decision tree is designed to reattach the words and affixes. The case of the first set of words is trivial, and they are reattached to the previous word. The second group of words makes use of another feature to clarify the context; an additional word vector, composed of the preceding and proceeding words. A sample record of the training data in this tree for word "می" is:

((Boolean attach_sign) ("غذا" (food)) ("ی") ("خورم" (I eat))).

The assumption has been made that compound verbs and nouns are always studied in a separate form (Azimizadeh and Arab, 2007).

3. Implementation

Festival software includes a part of speech tagger similar to the HMM-type taggers found in the Xerox tagger and others. (DeRose, 1988) In this method, first the words and their POS tags are extracted from training data and are then adjusted to the POS lexicon form. Each entry of the POS lexicon includes the word, probable POS tags and the probability distributions. After choosing the word and their probable tags the system subtracts their probability distributions with the N-gram values and based on this creates a list of candidates. In the final stage a Viterbi decoder chooses the POS tag candidate with the maximum probability.

3.1. The corpus

The corpus that used for training in this project is based on Orumchian's (Oroumchian et al., 2006) work. This data set is extracted from the BijanKhan's tagged corpus (BijanKhan, 2004), which is maintained at the Linguistics laboratory of the University of Tehran. The corpus is gathered from daily news and common texts. The first corpus contains 550 different tags. Since this tag-set is not suitable for training data and increases the error of the tagger, the set is reduced to 38 tags. There are 2,597,937 tagged words in the training data. Table 1 shows the tag-set distribution (Oroumchian et al., 2007).

TAG	Frequency	Probability
ADJ	22	8.46826E-06
ADJ_CMPR	7443	0.002864966
ADJ_INO	27196	0.010468306
ADJ_ORD	6592	0.002537398
ADJ_SIM	231151	0.088974829
ADJ_SUP	7343	0.002826473
ADV	1515	0.000583155
ADV_EXM	3191	0.001228282
ADV_I	2094	0.000806024
ADV_NEGG	1668	0.000642048
ADV_NI	21900	0.008429766
ADV_TIME	8427	0.003243728
AR	3493	0.001344528
CON	210292	0.080945766
DEFAULT	80	3.07937E-05
DELM	256595	0.098768754
DET	45898	0.017667095
IF	3122	0.001201723
INT	113	4.34961E-05
MORP	3027	0.001165155
MQUA	361	0.000138956
MS	261	0.000100464
N_PL	160419	0.061748611
N_SING	967546	0.372428585
NP	52	2.00159E-05
OH	283	0.000108933
P	319858	0.123119999
PP	880	0.00033873
PRO	61859	0.023810816
PS	333	0.000128179
QUA	15870	0.005934709
SPEC	3809	0.001466163
V_AUX	15870	0.006108693
V_IMP	1157	0.000445353
V_PA	80594	0.031022307
V_PRE	42495	0.01635721
V_PRS	51738	0.019915033
V_SUB	33820	0.013018022

Table 1: POS tags distribution

3.2. POS lexicon

This tagger has a POS lexicon that contains 61521 entries. To generate the POS lexicon the frequency of the each word, with an especial tag, is computed using the formula below:

$$P(Wi|Ti) = \frac{P(Ti|Wi)P(Wi)}{P(Ti)}$$

Finally, words and their POS candidates are written in the lexicon format, e.g.

("Amryka" nil ((N_SING 0.04816))); America

3.3 Language model

The language model type in this tagger is a trigram that is extracted by the Edinburgh Speech Tools (EST) (Black et al., 1998). This model contains 64003 trigrams. Linear interpolation method is used to smooth the zero values of the 3-grams. To extract this model with EST the training data should be changed into the input format of the *ngram_build*. There are two input formats: sentence per line, and N-gram per line.

The sentence per line format is useful for sliding-window type applications such as Automatic Speech Recognition (ASR) systems. The second type is also useful for non-sliding-window cases such as the discrete-time process. In this project the latter input format is used. (Black et al., 1998) An example record of the input file is:

prev_prev_tag prev_tag PREP N CONJ VDEC N N VLINK CONJ VDEC VDEC PUNC last_tag

The trigram file and POS lexicon are put in the parameters list of the EST Viterbi decoder. Viterbi decoder requires two functions at declaration time. The first constructs candidates at each stage, while the second combines paths. (Black et al., 2002)

4. Optimization

Because the training data includes incorrectly tagged words the POS lexicon will include noise. These can introduce errors in the decision making process of the system. Therefore to reduce the error rate, a linguistic specialist has manually corrected the lexicon.

Another source of error is abnormal trigram values. Some trigrams have very large values that cause the error in candidate choice process. These trigrams will always get high heuristic scores in the Viterbi decoder. For example, the trigram: N_SING N_SING N_SING has an abnormally large score of: 125,042; compare with the average trigram value of 20,000. To solve this problem a threshold value is set at 50,000. All trigram values above the threshold are clipped to the threshold value.

5. Evaluation

To evaluate the robustness of the system in different corpuses for TTS purposes, the tagger is evaluated in a variety of contexts such as humor, press reports, history and romance. This output of the algorithm is verified by a linguistic specialist. Each test data set contains 2000 words. Results of the system performance are listed in table 2.

The overall accuracy rate in context of romance is markedly different from others. This can be attributed to usage of rare, seldom encountered, literary terms in such texts. These terms don't observe the Part-of-Speech rules like other corpora and change the POS positions in the sentences. Therefore the corpus is not adopted with the language model and increases the error rate of the system.

The other important metric for the tagger is performance against the Out-of-Vocabulary (OOV) words. Table 3 outlines the results of evaluating this metric. System accuracy for known (lexical words) and unknown words is studied separately. Two conclusions are drawn from this table:

1. The size of unknown words versus the known words is very small because the lexicon that is used in this tagger is very large.
2. System accuracy for unknown words is very small. This is mainly due to unbalanced values of the n-grams in language model.

Balancing of n-gram values can be done in a number of ways, for example by using logarithmic values instead of direct values; more work is required to fully explore the solutions to this problem.

Context type	**Correct**	**% Accuracy**
Humor	1925	96.25
Press Reportage	1935	96.752
Learned	1905	95.27
Historic	1915	95.70
Romance	1832	91.58

Table 1: The Accuracy percentages in different contexts

Tagger	**Known word**	**Unknown word**	**known accuracy %**	**Unknown accuracy %**
Festival	18387	241	96.136	60.25

Table 2: The POS tagger benchmark

6. Conclusion

In this paper the implementation of a Persian POS tagger based on HMM, is tested and an optimization process is suggested. There is little deviation in the accuracy rate of the tagger, with the exception of corpus collected from romance. As this discussed, this is mainly due to usage of uncommon words in such bodies of work. The system is robust in different contexts and can be used as the common model.

Using the HMM model, the accuracy rate of the system is high and quite comparable with best-case results obtained in other languages. The performance of the system is higher than that of the work done by Assi and Jabbari and approximately the same as the Orumchian's tagger. Although because of the different testing data this comparison is not accurate.

The other advantage of this approach is implementation in festival software that it makes system much extendable. This software provides especial facilities such as Classification and Regression Tree (CART), Viterbi Decoder and Scheme Interpreter that help users to develop their systems in other approaches. In this software several NLP processes such as Persian Morphological Parser (Azimizadeh and Arab, 2007) and Semantic Parser are used that aid the tagger in resolving erroneous tags. This subject will be studied more in depth in the ParsGooyan TTS system and we are going to obtain the best results of Persian NLP by providing a multi- approach system.

Acknowledgements

We would like to thank Dr. Orumchian for allowing access to his training data. We would also like to thank Dr. Alan Black for his useful guidance and to Arvin Farahman for reviewing the paper.

References

Assi M. and Haji Abdolhossini M. (2000). Grammatical Tagging of Persian Corpus. *International Journal of Corpus Linguistics*. 5(1), pp. 69-82.

Azimizadeh A. and Arab M. M. (2007). The Persian Morphological parser by Using POS Tagger. In *CAASL-2 Proceedings.* Stanford university, pp.22-29.

BijanKhan M. (2004). The Role of the Corpus in Writing a Grammar: An Introduction to a Software. *Iranian Journal of Linguistics*, Vol. 19, no. 2.

Black A W. and Taylor P. A (1997). Assigning Phrase Breaks from Part-of-Speech Sequences. *Eurospeech97*. Rhodes, Greece.

Black A W., Taylor P. A. and Caley R. (2002). *The Festival speech synthesis system, Version 1.4.3.* pp.73-75, http://www.cstr.ed.ac.uk/projects/festival/manual.

Brants T. (2000). TnT – a Statistical Part-of-Speech Tagger. In *Proceedings Sixth conference on applied natural language processing (ANLP).* Seattle, WA.

Brill E. (1995). Transformation-Based Error-Driven Learning and Natural Language Processing: A case Study in Part of Speech Tagging. *Computational Linguistics*.

DeRose S. (1988).Grammatical category disambiguation by statistical optimization. *Computational Linguistics,* 14.

Hepple M. (2000). Independence and Commitment: Assumptions for Rapid Training and Execution of Rule-based Part-of- Speech Taggers. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*. Hong Kong.

Jabbari S. and Allison B. (2007). Persian Part of Speech Tagging. In *CAASL-2 Proceedings.* Stanford University, pp. 67-74.

Jurafsky D. and Martin J. H. (1999). Speech and language Processing. *Prentice Hall*, September 28, pp. 283-443.

Oroumchian F., Tasharofi S., Raja F. and Rahgozar M. (2007). Evaluation Of Statistical Part Of Speech Tagging Of Persian Text. *International Symposium on Signal processing and its application.* Sharjah, United Arab Emirates.

Oroumchian F., Tasharofi S., Amiri H., Hojjat H. and Raja F. (2006). Creating a Feasible Corpus for Persian POS Tagging. *UOWD Technical Report Series*.

Padr´o M. and Padr´o L. (2004). Developing Competitive HMM POS Taggers Using Small Training Corpora. *Estal*.

Taylor P., Caley R. and Black A. W. (1998). *The Edinburgh Speech Tools Library*. The Centre for Speech Technology Research, University of Edinburgh, 1.0.1 edition. http://www.cstr.ed.ac.uk/projects/speechtools.html.