ONOMÁZEIN
Journal of linguistics, philology and translation

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
FACULTAD DE LETRAS

An intelligent extension of the training set for the Persian n-gram language model: an enrichment algorithm

Rezvan Motavallian
University of Isfahan
Iran

Masoud Komeily
University of Isfahan
Iran

ONOMÁZEIN 61 (September 2023): 191-211
DOI: 10.7764/onomazein.61.09
ISSN: 0718-5758

Rezvan Motavallian: Linguistics Department, Faculty of Foreign Languages, University of Isfahan, Iran. Orcid: 0000-0002-2319-8414. | E-mail: r.motavallian@fgn.ui.ac.ir / rezvan.motavallian@gmail.com
Masoud Komeily: Linguistics Department, Faculty of Foreign Languages, University of Isfahan, Iran. | E-mail: masoud.komeily@fgn.ui.ac.ir

Received: March 2020
Accepted: December 2020

Abstract

In this article, we are going to introduce an automatic mechanism to intelligently extend the training set to improve the n-gram language model of Persian. Given the free word-order property in Persian, our enrichment algorithm diversifies n-gram combinations in baseline training data through dependency reordering, adding permissible sentences and filtering ungrammatical sentences using a hybrid empirical (heuristic) and linguistic approach. Experiments performed on baseline training set (taken from a standard Persian corpus) and the resulting enriched training set indicate a declining trend in *average relative perplexity* (between 34% to 73%) for informal/spoken vs. formal/written Persian test data.

Keywords: training corpus; n-gram language model; dependency parsing; enrichment algorithm; free word-order.

1. Introduction

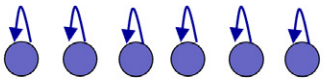
Language modeling is a fundamental task in computational linguistics and natural language processing playing a crucial role for text-disambiguation in statistical machine translation (SMT) for text-to-text conversion, automatic speech recognition (ASR) for speech-to-text conversion, optical character recognition (OCR) for image-to-text conversion, and in spell checkers to reduce grammatical errors through text-to-text conversion (Clark et al., 2010: 74; Koehn, 2010: 181; Jurafsky and Martin, 2009: 83).

The n-gram model is the most common language modeling technique employing statistical methods and Markov chains to predict the next word in a given sentence by learning from huge amounts of training data (corpus) already written by human. In 1-gram model, the probability of each word is estimated by conditioning on the word itself, independently from all previous words. In 2-gram and 3-gram models, however, each word is conditioned on the previous word and the two previous words respectively (see Figure 1).

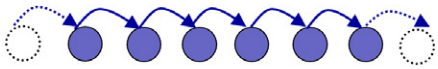
FIGURE 1

Probability of a sentence in 1-gram, 2-gram, and 3-gram language models

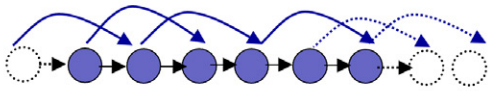
$$P(S) = \prod_{i=1}^{n} P(w_i|w_1 \dots w_{i-1}) \approx \prod_{i=1}^{n} P(w_i)$$



$$P(S) = \prod_{i=1}^{n} P(w_i|w_1 \dots w_{i-1}) \approx \prod_{i=1}^{n} P(w_i|w_{i-1})$$



$$P(S) = \prod_{i=1}^{n} P(w_i|w_1 \dots w_{i-1}) \approx \prod_{i=1}^{n} P(w_i|w_{i-2}w_{i-1})$$



This simplification in practice ignores long distance dependency among words in a sentence. Moreover, the automatic learning of language model for all languages including Persian suffers from insufficient and sparse training data due to inherent productivity encoded in natural language—no training data of any size could ever include all words, sentences, and linguistic rules. As a result, many 1-gram, 2-gram, and 3-gram word combinations in a new sentence (test data) may not have been seen in training data, leading to a zero or near-zero probability for the sentence—a problem commonly dealt with while using various smoothing techniques (Heafield et al., 2013; Nugues, 2014: 140).

Persian language is an Indo-European language spoken by around 110 million people primarily in Iran, Afghanistan, and Tajikistan (Windfuhr, 2009). It uses Perso-Arabic script with a right-to-left writing direction and is considered as a language with a canonical subject-object-verb (SOV) word-order (Khanlari, 1995; Dabir-Moghadam, 2013). However, Persian enjoys a relatively flexible word-order especially in the colloquial spoken domain, making it a free word-order language along with German, Russian, Turkish, Indian, Japanese, and Czech (Mahajan, 1995; Bailyn, 1999; Karimi, 1999; Sabel and Saito, 2005: 3; Carnie et al., 2014: 265).

Utilizing the free-word-order phenomenon in Persian, especially in the colloquial and informal domain, in this paper, a new algorithm is introduced for the first time to improve the Persian n-gram language model by an intelligent and automatic extension of training data. In our enrichment algorithm, to cope with data sparseness and improve the Persian n-gram model, permissible sentence permutations are automatically produced through dependency parsing while considering word-order criteria in Persian combined with handcraft empirical and computational heuristics.

Our main motivation to use dependency parsing in our enrichment algorithm is that dependency formalism is more similar to the human understanding of language and can easily represent the free word-order nature of syntactic roles in sentences (Kübler et al., 2009). Furthermore, the only treebank currently suitable for analyzing Persian sentences is the Persian Dependency Treebank, which includes approximately 30,000 sentences from contemporary Persian-language texts manually tokenized and annotated at morphological and syntactic levels, containing nearly 4500 distinct verb lemmas (Rasooli et al., 2013; Rasooli et al., 2011).

What follows is organized as below: in section 2, related works on Persian language modeling are briefly reviewed. In section 3, the three phases of enrichment algorithm, its linguistic and computational bases, and word-order criteria in Persian are elaborated. The experimental setup is explained in section 4, and the conclusion and future works are finally given in section 5.

2. Related works

Computational models of Persian language are not numerous. Regarding statistical approaches, there are only a few works to mention, all of which primarily use class-based methods to deal with data sparseness in Persian n-gram language modeling.

Bahrani et al. (2006) have constructed a class-based and POS-based n-gram language model (Brown et al., 1992) using Bijankhan corpus (Bijankhan, 2004). Bazargani and Almasganj (2007) used the class-based n-gram language model for a limited news wire Persian text combining Brown and Martin algorithm (Brown et al., 1992; Martin et al., 1998)

and word similarity measure. Bahrani and Sameti (2011) and Bahrani et al. (2011) report implementation results of various rule-based and statistical language models for continuous speech recognition (CSR).

Among rule-based (linguistic) approaches, Bahrani et al. (2011) developed an extension to Hafezi-Manshadi's (2001) work, building upon the ideas presented by Meshkatoddini (2003). This work is characterized by a large-coverage grammar with a vocabulary size of 1200 words tailored for continuous speech recognition (CSR) applications in Persian. Using a bottom-up chart parser (Allen, 1995), Bahrani et al. (2011) report an acceptance rate of 89%. Also, combining their rule-based language model with a 2-gram statistical language model, word error rate (WER) in a Persian continuous speech recognition system dropped by about 31%. Based on Generalized Phrase Structure Grammar (GPSG) (Gazdar et al., 1985), their rule-based language model describes the Persian grammatical structure by means of a set of 121 rules covering major syntactic rules of modern Persian. Their grammar, however, cannot properly describe coordination (*i.e.* 'دو یا سه کتاب' [do yɑ se ketɑb] 'two or three books'), becomes problematic with respect to separable complex verbs (i.e. 'او درد را احساس می کند' [u dærd rɑ ehsɑs mi-konæd] 'he/she feels the pain' vs. 'او احساس درد می کند' [u ehsɑs-e dærd mi-konæd] 'he/she feels the pain'), and cannot handle words containing copula (the enclitic form of the verb 'بودن' [budæn] 'to be' in the present indicative) as in 'مشغولند' [mæshqul-ænd] '(they) are busy'.

Dehdari and Lonsdale (2008) is a link grammar parser based on Sleator and Temperley (1991). Furthermore, Valad (2006) employs a unification-based approach to Persian grammar utilizing semantic features for Persian syntactic disambiguation. Ayat (2001) is a head-driven phrase structure grammar (HSPG) for Persian based on Pollard and Sag (1994).

Our contributions are as follows:

i. Considering free word-order property of Persian: Unlike previous works on the n-gram language modeling of Persian, our proposed algorithm takes into account the free word-order property of Persian; it also offers an automatic approach to deal with the infamous data sparseness problem in modeling all natural languages highly aggravated by a free word-order phenomenon in free word-order languages like Persian. In our research, we mainly focus on the artificial extension of the input of the language modeling, i.e. training data, as much as possible. To the best of our knowledge, this approach has never been taken in modeling any language in general and Persian (or other free word-order languages) in particular.

ii. Intelligent extension of training data: Our enrichment algorithm (section 3) provides an intelligent mechanism to automatically expand training data especially for the colloquial and informal domain to be used for Persian continuous speech recognition (CSR). The importance of such an extension stems from the fact that

creating text corpora is essentially a time-consuming task. Furthermore, the creation of spoken text corpora via the manual conversion of audio files into text is cumbersome by itself. More importantly, there are not enough spoken corpora currently available in Persian.

iii. Reference data for word-order in Persian: The automatic generation of permutations in our enrichment algorithm provides a fundamental infrastructure for Persian NLP research associated with word-order in tasks like statistical machine translation (SMT) (e.g. pre-reordering and reordering). Considering the free word-order in Persian, Persian currently lacks a reference word-order for SMT research.

3. Enrichment algorithm

In this section, we will describe three phases of our training set enrichment algorithm together with related steps within each phase. The algorithm is aimed at an intelligent and automatic expansion of training data to diversify n-gram combinations in the baseline training set, considering the free word-order nature of Persian language especially in informal speech and colloquial genre.

The general diagram of the algorithm is shown in Figure 2. There are three separate phases: phase I (preprocessing), phase II (dependency parsing), and phase III (permutation generation/filtration). Many steps were taken in designing and testing the algorithm through linguistic rules combined with computational heuristics.

3.1. Preprocessing

In the first phase, four preprocessing steps were taken:

i. Normalization: Due to numerous orthographical inconsistencies in Persian (Shamsfard, 2011), the input text was initially normalized.

ii. Tokenization: multiword expressions (MWEs) such as multitoken verbs were tokenized and *frozen* to act as one-word expressions (e.g. 'خواهم رفت' [xɑhæm ræft] 'I will go' is converted into 'خواهم_رفت');

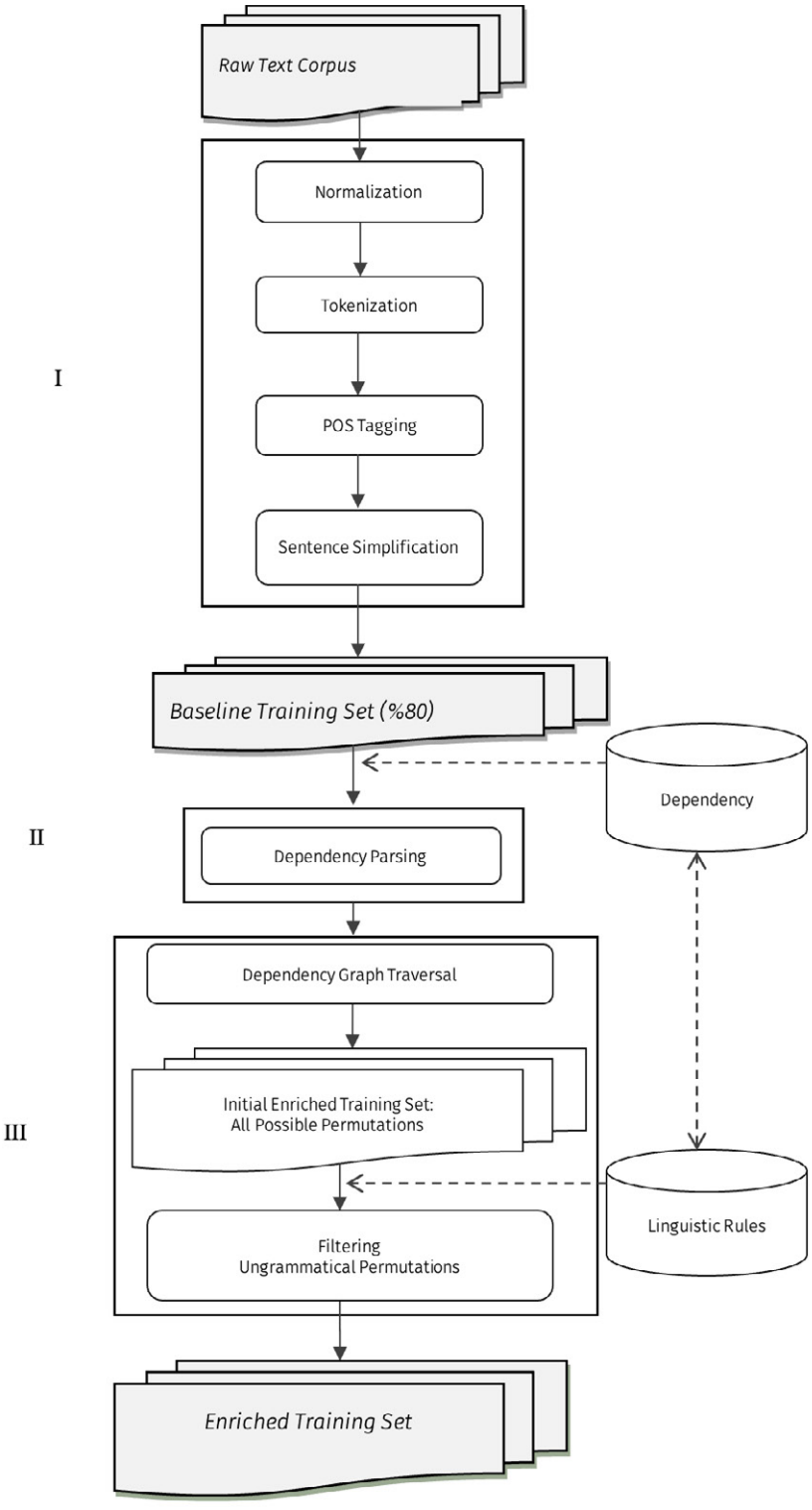
iii. POS tagging: required for sentence simplification and dependency parsing.

iv. Sentence simplification: Complex sentences were broken into simple sentences through POS-tag heuristics based on the order of verb (V), punctuation (PUNC), and conjunction (CONJ)[1].

1 Conjunctions in Persian include 'که' [ke] 'that', 'و' [væ] 'and', etc.

FIGURE 2

Diagram of training set enrichment algorithm including three phases shown as three separate boxes



Dependency grammar is commonly attributed with weakness in describing complex sentences, i.e. sentences with more than one verb, as there is ambiguity in determining the true root of the sentence—hence the existence of at least two possible parse trees (Schneider, 1998; Bach, 2012).

The sentence simplification therefore proves beneficial to fine-tune dependency parsing in the second phase of the algorithm and, as a result, will enhance the overall performance of our enrichment algorithm.

3.2. Dependency parsing

In phase II (dependency parsing), 80% of sentences already preprocessed in phase I were randomly selected which will serve as the baseline training set in language modeling (section 4). The resulting preprocessed text corpus was then automatically parsed after training MatlParser[2] with Persian Dependency Treebank.

3.3. Permutation generation/filtration

In this part, after describing the three steps taken in phase III, we will justify both the computational and the linguistic basis of our enrichment algorithm using word-order criteria in Persian.

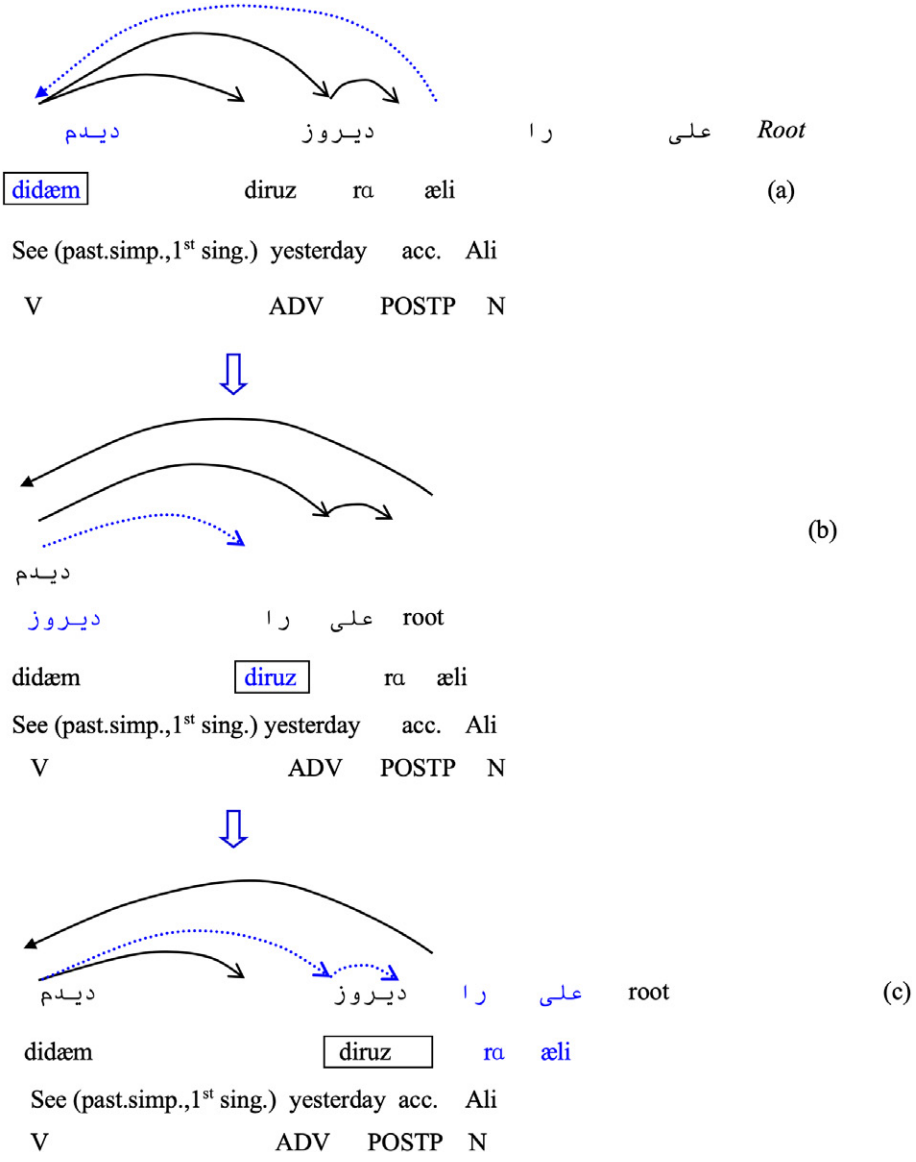
The resulting labeled directed graph (LDG) associated with each sentence is then traversed to extract obligatory and optional complements associated with the verb in each sentence. Extracted parts are permuted afterwards producing all possible grammatical reordering (permutations) for each Persian sentence. The valency structure of the verb in dependency grammar includes both the obligatory complements (OBC) and the optional complements (OPC) a verb can obtain. Unlike OPC, the removal of OBC from the head of a syntactic phrase leads to a semantically ill-formed phrase (Tabibzadeh, 2012).

One should bear in mind that since finding the correct word-order in natural languages is an NP-complete problem (Knight, 1999), approximation and heuristic rules need to be employed in phase III. Therefore, after extensive empirical investigations, three handcraft heuristics were implemented in phase III to reduce search space as much as possible and block/filter ungrammatical permutations: 1) no permutation for original sentences containing punctuation (PUNC), 2) no permutation for the original sentences with length below 6 (i.e. sentences for which the permutation length exceeds that of the original sentence due to errors in POS tagging or parsing), and 3) no permutation for sentences still containing more than one verb (ROOT) due to normalization, tokenization, or POS tagging errors in phase I.

2 www.maltparser.org

FIGURE 3

Extracting obligatory and optional complements by traversing its LDG. Dependency relations are shown over each arc. 'را' [rɑ] is direct object (accusative) marker in Persian



Although Persian is commonly known as a subject-object-verb (SOV) language, there is a great deal of flexibility in Persian especially in the colloquial and informal spoken domain. For example, while the English sentence 'I saw Ali yesterday' has only two permissible (grammatical) permutations (i.e. 'I saw Ali yesterday', 'Yesterday I saw Ali'), its corresponding Persian equivalent has 3!=6 possible permutations. Figure depicts labeled directed graph (LDG) for the above sentence with various dependency relations between the central head, i.e. 'دیدم' [didæm] 'see' (past simple, 1st sing) and its direct/indirect dependents.

By traversing the graph, obligatory/optional complements of the verb 'دیدن' ['didæn'] 'to see' are extracted in three stages.

To linguistically justify the computational basis of our approach in phase III, we employed Dabir-Moghadam (2013), which is an attempt to introduce 24 'correlation pairs', 'components', or 'criteria' for word-order in Persian based on Dryer's (1992) comprehensive typological study on many languages. As seen in Table 1, each of those word-order criteria can be associated with one or more dependency relations defined in Persian Dependency Treebank.

TABLE 1

Word-order criteria in Persian vs. dependency relations in Persian Dependency Treebank[3]

	WORD-ORDER CRITERION	DEPENDENCY RELATION
1	Preposition + N / N + Postposition[4]	NPP
2	Numerator + N	MESU
3	N + Indefinite article [-i] / N + Definite article [-e]	*
4	N + Possessive free morpheme	*
5	Ezafe construction[5]	MOZ
6	Demonstrative Adj + N	NPREMOD
7	Adjectival ezafe construction (Modified + Modifier)	NEZ
8	Quantity Adverb + Adj	APREMOD
9	Comparative Adj + PP / PP + Comparative Adj	COMPPP
10	PP + V	VPP
11	Manner Adv + V	ADV

3 Nine valency complements are defined in Persian Dependency Treebank (PerDT) in addition to 18, 11, 7, and 7 dependency relations for verb, noun, adjective, and other parts of speech respectively (Rasooli et al., 2014).

4 As the only postposition used in Persian today, 'را' [ra] is a direct object marker.

5 Ezafe is an unstressed vowel -e that occurs at the end of some words (-ye on some specific occasions) that links together elements belonging to a single constituent (Ghomeshi, 1997). The first word is called Mozaf (s.th. which is added to s.th. else), the second word Mozafon-elayh (s.th. to which s.th. else is added). It approximately corresponds in usage to the English preposition "of", joining two nouns (possessive ezafe) (e.g. 'کتابِ من' [ketɑb-e mæn] 'my book') or nouns and adjectives (adjectival ezafe) (e.g. 'مردِ جوان' [mærd-e jævɑn] 'young man') (Abrahams, 2004), or connecting parts of a name (i.e. first and last name) or title (e.g. 'احمدِ علوی' [æhmæd-e ælævi] 'Ahmad Alavi') (Nourian et al., 2015).

12	Noun Head + 'که'[6] + Relative cl.	VCL
13	Main clause + 'که' + subordinate clause / complement clause	AJUCL / VCL
14	Adverbial clause subordinator / adverbial conj. + 'که' + clause	AJUCL / PRD
15	Verb 'خواستن' [xɑstæn] 'to want' + 'که' + subordinate clause v.	VCL / PRD
16	Auxiliary verb + main verb / main verb + auxiliary verb	-
17	Aux. verb 'توانستن' [tavɑnestæn] 'to want' + main verb	-
18	Negative prefix + verb stem	*
19	Tense/aspect prefix + verb stem / verb root + tense/aspect postfix	*
20	Question Particle + sentence	PART
21	WH word position	PART / MOZ
22	Subj + pred + predictive verb	MOS
23	Subject + verb	SBJ
24	Obj (NP) + verb / verb + obj (clause)	OBJ / VCL

Correspondence between 24 word-order criteria in Persian (Dabir-Moghadam, 2013) and dependency relations defined in Persian Dependency Treebank (Rasooli et al., 2013). Numbers 3, 4, 18, and 19 are morphological, and thus not relevant to our syntactic reordering. Numbers 16 and 17 are manually preserved within the 'filtering' step of our enrichment algorithm.

Among them, however, word-order criteria no. 3, 4, 18, 19 are merely morphological, playing no role in syntactic parsing in our algorithm. Also, word-order criteria for Persian auxiliary/main verb (no. 16 and 17) were manually encoded in the 'filtering' step of our algorithm.

Finally, Karimi-Doostan's (2011) criteria for the separability of Persian complex verbs were included as part of additional heuristic rules to the further blocking of ungrammatical permutations, given the productive abundance of light verb constructions in Persian language (Samvellian and Faghiri, 2013).

4. Experimental setup

We now evaluate our *enrichment* algorithm and investigate its effectiveness in improving the Persian n-gram model in informal/spoken domain. We will first describe data and tools used in our experiments and then discuss the results and the analysis.

6 'که' 'ke' here functions as a relative pronoun in Persian, equivalent to 'that' (who, which, whose, whom) in English.

4.1. Data and tools

i. Data: We adopt part of the Bijankhan corpus (Bijankhan, 2004) as our baseline training set to implement our *enrichment* algorithm and produce an enriched training set. The Bijankhan corpus is a tagged corpus suitable for Persian natural language processing gathered from daily news and common texts; the free version[7] has about 2.6-million manually tagged words with a tag set of 40 POS tags.

   For evaluation purposes, we consider two types of test data: 1) formal/written test sets created by using the Bijankhan corpus, Soltanzadeh (2013)[8], and IRNA[9] news agency; 2) informal/spoken test sets constructed by the use of the Persian section of Tehran English-Persian parallel corpus (TEP)[10] (Pilehvar et al., 2011).

ii. Hazm: All preprocessing steps in phase I of enrichment algorithm (i.e. normalization, tokenization, and breaking down complex sentences into simple sentences) were done using Hazm[11] package for Persian language processing in Python.

iii. POS tagging: Stanford POS tagger[12] already trained with the Bijankhan (2004) 10-million-word annotated corpus was used for POS tagging.

iv. Parsing: We trained java-based MaltParser[13] (V.1.8.1) with Persian Dependency Treebank (PerDT)[14] (V.1.1.1) (Rasooli et al., 2013) for dependency parsing in phase II of our algorithm. The treebank contains around 30,000 syntactically and morphologically annotated Persian sentences within the framework of dependency grammar.

v. Permutation generation/filtration: Phase III of our *enrichment* algorithm was implemented after extensive empirical experiments utilizing the Persian verb valency lexicon (Per-Vallex)[15] (V.3.0) (Rasooli et al., 2011) as a valuable resource including obligatory and optional complements of nearly 4500 distinct verb lemmas of simple, complex, prefixed, and phrasal Persian verbs.

---

7 http://ece.ut.ac.ir/dbrg/bijankhan/

8 The data used with permission had been originally used in automatic transformation of Persian dependency trees to their corresponding phrase structure trees.

9 www.irna.ir

10 TEP corpus, as a collection of Persian subtitles of English movies, is available at http://opus.lingfil.uu.se/TEP.php.

11 http://www.sobhe.ir/hazm/

12 http://nlp.stanford.edu/software/tagger.shtml

13 http://www.maltparser.org

14 http://dadegan.ir/catalog/perdt

15 http://dadegan.ir/catalog/pervallex

vi. N-gram modeling: We used KenLM[16] (Heafield et al., 2013) in the Linux platform, a fast, lightweight, and scalable set of programs written in C++ to build the n-gram model and estimate n-gram probabilities for baseline and enriched training sets.

4.2. Results and analysis

Now we present the experimental results of implementing our *enrichment* algorithm on a randomly selected section of the Bijankhan corpus. After preprocessing (phase I), we randomly selected 80% of the sentences as our baseline training set, which were then used as our input for phase II (dependency parsing) and then for phase III (permutation generation/filtration) of the *enrichment* algorithm.

Table 2 summarizes the total words, number of sentences, and n-gram counts of the baseline training set and the resulting enriched training set. As evident, word-types (unigrams) are equal in both baseline and enriched training sets while the number of 2/3/4-grams have all increased as a result of the *enrichment* algorithm. In other words, we now have two training sets with equal vocabulary size (1-gram), but different and diversified 2/3/4-gram combinations (i.e. word orders) in the enriched training set—highly valuable for building a language model for a free-word-order language like Persian.

TABLE 2

Language model: baseline vs. enriched training set

	BASELINE TRAINING SET	ENRICHED TRAINING SET	INCREASE (%)
Total words	619,310	2,410,000	289,14
Sentences	67,279	194,258	188,73
1-gram	34,800	34,800	00,00
2-gram	291,260	341,217	17,15
3-gram	479,245	655,894	36,86
4-gram	509,976	775,436	52,05

N-gram language model statistics for baseline vs. enriched training sets. Both training sets have an equal vocabulary size (unigram count).

To evaluate our enrichment algorithm and investigate the effectiveness of the enriched vs. baseline training set in improving the Persian n-gram model, we prepared various Persian

16 https://kheafield.com/code/kenlm/

test data in different domains. In addition to BIJ-1 (the remaining 20% of our originally selected corpus), we also prepared 6 other test sets. Table 3 summarizes all 7 test sets ordered with an increasing size in two different domains: formal (written) and informal (spoken).

TABLE 3

Test sets for evaluating training set enrichment algorithm

TEST SET	SOURCE	SIZE (K)	DOMAIN
IRN	IRNA news	3.8	Formal
SOL	Soltanzadeh (2013)	3.8	Formal
BIJ-2	Bijankhan	10	Formal
BIJ-1	Bijankhan	163	Formal
TEP-1	TEP	28	Informal
TEP-2	TEP	47	Informal
TEP-3	TEP	65	Informal

Formal (written) and informal (spoken) test sets are increasing in size.

For the evaluation metric, we use *perplexity (pp)* as a standard measure of language model quality as below:

$$PP = 2^{-ALP} \tag{1}$$

in which the average log probability (ALP) is defined as:

$$ALP = \frac{\log_2 \prod_{i=1}^{m} P(x^{(i)})}{M} \tag{2}$$

where there are m sentences $x^{(1)}$, $x^{(2)}$,...$x^{(m)}$ with a total number of M words in test data. Perplexity is in fact a measure showing how many sequences of words in test data are correctly predicted using n-gram probabilities already obtained from training data (Jurasfky and Martin, 2009); the lower the perplexity, the better the language model is.

In our experiments, n-gram probabilities (1-gram to 4-gram) were computed via KenLM (Heafield et al., 2013) with the Modified Kneser-Ney (MKN) smoothing technique (Kneser and Ney, 1995). Table 4 shows perplexity (henceforth called *absolute perplexity*) for various test sets computed via n-gram probabilities estimated out of the baseline training set (BL2, BL3,

BL4) and the enriched training set (EN2, EN3, EN4) for 2-gram, 3-gram, and 4-gram respectively. Also seen in this table is perplexity difference, hereinafter called *relative perplexity* (Diff), along with *average relative perplexity* (Avg) for formal (F) and informal (I) test sets collectively (e.g. Diff=EN2-BL2=251 is equal to +34.91% increase in *relative perplexity*).

TABLE 4

Absolute perplexity with baseline (BL) and enriched training set (EN), relative perplexity (Diff), and the average relative perplexity (Avg) for formal (F) and informal (I) test sets

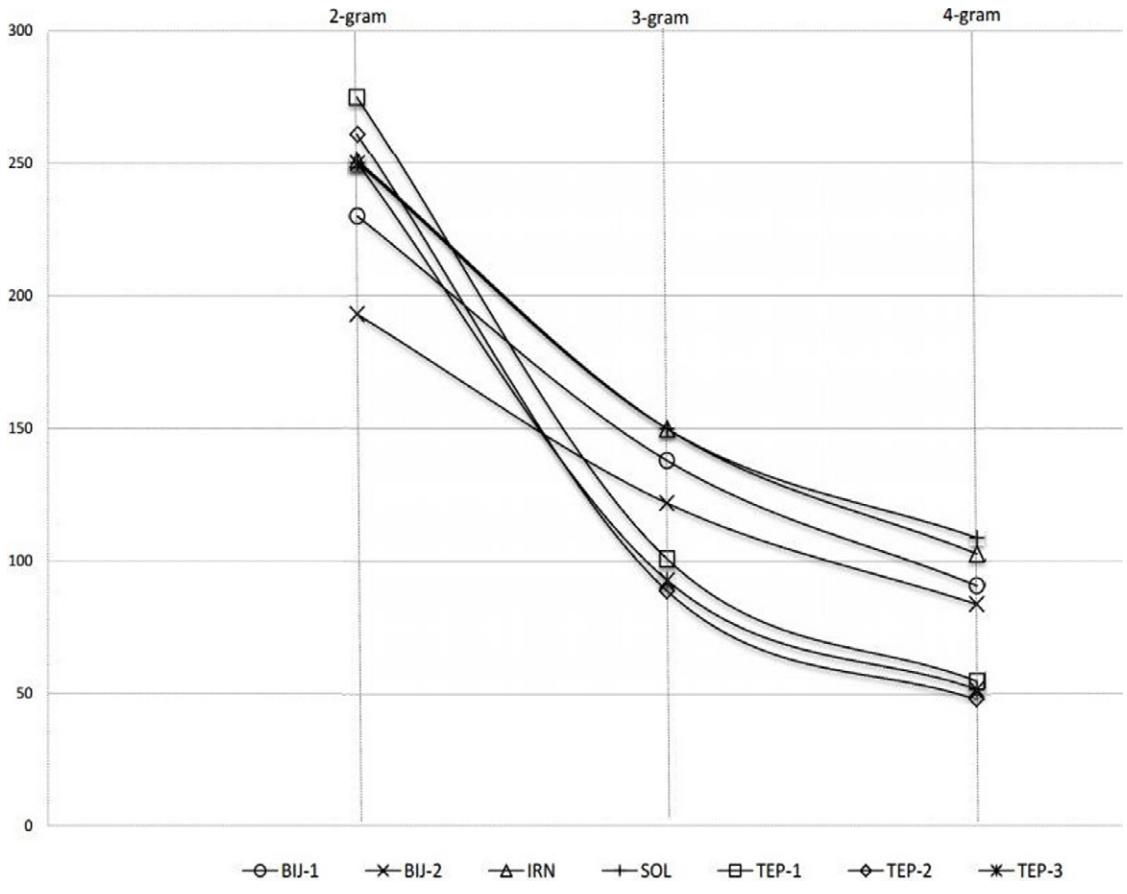
Test Set Size(K)	BL2	EN2	Diff (%)	Avg	BL3	EN3	Diff (%)	Avg	BL4	EN4	Diff (%)	Avg
IRN (F) 3.8	468	719	251(+34.91)		396	546	150(+27.47)		389	492	103(+20.93)	
SOL (F) 3.8	492	742	250(+33.69)		431	581	150(+25.82)		424	533	109(+20.45)	
BIJ-2 (F) 10	341	534	193(+36.14)	+35.11	285	407	122(+29.98)	+27.93	282	366	84 (+22.95)	+21.35
BIJ-1 (F) 163	414	644	230(+35.71)		347	485	138(+28.45)		341	432	91 (+21.06)	
TEP-1 (I) 28	924	1199	275(+22.94)		910	1011	101 (+9.99)		904	959	55 (+5.74)	
TEP-2 (I) 47	921	1182	261(+22.08)	+23.01	898	987	89 (+9.02)	+9.88	892	940	48 (+5.11)	+5.70
TEP-3 (I) 65	791	1041	250(+24.02)		781	874	93 (+10.64)		777	829	52 (+6.27)	
				34.46				64.61				73.28

As seen, the *relative perplexity* (Diff) for all test sets on all n-gram levels (n=2, 3, 4) is positive—an apparently discouraging result, since we expected to obtain lower perplexities with the enriched training set in comparison with the baseline training set! Nevertheless, this undesired increase slows down as we move to informal test sets (e.g. from +35.11% down to +23.01% for 2-gram). Moreover, this declining growth rate grows much faster when moving from 2-gram to 4-gram: the reduction of *average relative perplexity* (Avg) from formal test sets to informal test sets is -34.46%, -64.61%, and -73.28% respectively (see also Figure 4). Therefore, we are convinced that this reduction is convincing enough to validate the effectiveness of our *enrichment* algorithm to enrich/diversify n-gram combinations (n>1) for the purpose of improving the Persian language model in the informal/spoken domain.

Absolute perplexity values for baseline (BL) and enriched (EN) training sets computed for various formal (F) and informal (I) test sets. Perplexity difference or *relative perplexity* (Diff) is computed for each test set. A falling trend is visible for the *average relative perplexity* (Avg) while moving from formal (F) to informal (I) test sets on the one hand, and from 2-gram to 4-gram on the other hand.

FIGURE 4

Faster decline rate of *relative perplexity* for informal/spoken test data (TEP-1, TEP-2, TEP-3) vis-à-vis formal written test data (IRN, SOL, BIJ-2, BIJ-1), and from 2-gram to 4-gram



5. Conclusion

In this paper, utilizing combinatory linguistic and computational approaches, and considering the free-word-order phenomenon in Persian language, we introduced our *enrichment* algorithm to artificially extend Persian training data and intelligently enrich/diversify n-gram combinations through dependency reordering. Our primary goal was to help improve the Persian n-gram language model in the colloquial informal domain. However, con-

sidering the fundamental importance of language models in computational linguistics and natural language processing for tasks like statistical machine translation (SMT), automatic speech recognition (ASR), and optical character recognition (OCR), the resulting n-gram model based on the enriched training set can be externally evaluated in the aforementioned applications. Therefore, based on the obtained results in part 4.2, for future works, it seems quite motivating to externally test the language model resulting from enriched training set for Persian speech recognition.

Since word-order relaxation poses many challenges to Persian language processing, especially in SMT, the subsequent Persian language model can be tested in statistical machine translation (SMT), e.g. in English-to-Persian SMT[17]. Finally, from a linguistic and typological point of view, our *enrichment* algorithm can be tailored and applied to other free-word-order languages like German, Russian, Indian, Turkish, Japanese, etc.

6. Bibliographic references

Abrahams, Simin, 2004: *Modern Persian: a course book*, Routledge, USA.

Allen, James, 1995: *Natural Language Understanding*, Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

Ayat, M. 2001: *A Computational Grammar for Persian Language*. Master thesis, AmirKabir University of Technology, Iran. (In Persian).

Bach, Nguyen, 2012: *Dependency Structures for Statistical Machine Translation*. Ph.D. thesis, Language Technologies Institute, School of Computer Science, Pittsburgh, PA, USA, Carnegie Mellon University.

Bailyn, John F., 1999: "On Scrambling: a reply to Boskovic and Takahasi", *Linguistic Inquiry* 30, 825-831.

Bahrani, Mohammad, Hossein Sameti, & Mehdi Hafezi-Manshadi, 2006: "A computational grammar for Persian based on GPSG", paper presented in the *2nd Workshop on Persian Language and Computer*, Tehran. (In Persian).

Bahrani, Mohammad, & Hossein Sameti, 2011: "Building statistical language models for Persian continuous speech recognition systems using the peykare corpus", *Intern J Comp Process Lang* 23 (1), 1-20.

17 To our knowledge, Persian SMT currently lacks reference word-order data.

Bahrani, Mohammad, Hossein Sameti, & Mehdi Hafezi-Manshadi, 2011: "A Computational grammar for Persian based on GPSG", *Language Resoureces and Evaluation* 45 (4), 387-407.

Bazargani, Negar, & Farshad Almasganj, 2007: "Word clustering for Persian n-gram language model", *Pardazesh-e-ala'em va dade'ha (Data and Signal Processing)* 4, 80-91. (In Persian).

Bijankhan, Mahmood, 2004: "The role of the corpus in writing a grammar: an introduction to a soft- ware", *Iranian Journal of Linguistics* 19 (2), 48-67. (In Persian).

Brown, Peter F., Vincent J. della Pietra, Peter V. de Souza, Jenifer C. Lai, & Robert L. Mercer, 1992: "Class-Based n-gram Models of Natural Language", *Computational Linguistics* 18, 467-479.

Carnie, Andrew, Yosuke Sato, & Daniel Siddiqi (eds.), 2014: *The Routledge Handbook of Syntax*, New York: Routledge.

Clark, Alexander, Chris Fox, & Shalom Lappin (eds.), 2010: *The Handbook of Computational Linguistics and Natural Language Processing,* USA: Wiley-Blackwell.

Dabir-Moghadam, Mohammad, 2013: *Typology of Iranian languages*, vol 2, Tehran: SAMT. (In Persian).

Dehdari, Jon., & Deryle Lonsdale, 2008: "A link grammar parser for Persian" in Simin Karimi, Donald Stilo, & Vida Samiian (eds.): *Aspects of Iranian linguistics*, vol. 1, Cambridge: Cambridge Scholars Press.

Dryer, Matthew S., 1992: "The Greenbergian word order correlations", *Language* 68, 81-138.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, & Ivan A. Sag, 1985: *Generalized Phrase Structure Grammar*, Oxford: Blackwell, and Cambridge, MA: Harvard University Press.

Ghomeshi, Jila, 1997: "Non-projecting nouns and the ezafe: Construction in Persian", *Natural Language & Linguistic Theory* 15 (4), 729-788.

Hafezi-Manshadi, Mehdi, 2001: *Design and Implementation of a Syntactic Parser for Persian Written sentences*. Master thesis, Sharif University of Technology, Iran. (In Persian).

Heafield, Kenneth, Ivan Pouzyrevsky, Jonathan H. Clark, & Philipp Koehn, 2013: "Scalable Modifed Kneser-Ney Language Model Estimation" in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 2: *Short Papers*, Sofia, Bulgaria: Association for Computational Linguistics, 690-696.

Jurafsky, Daniel, James H. Martin, 2009: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Upper Saddle River, New Jersey: Pearson, Prentice Hall.

KARIMI–DOOSTAN, Gholamhossein, 2011: "Separability of light verb constructions in Persian", *Studia Linguistica* 65 (1), 70-95.

KARIMI, Simin, 1999: "Is scrambling as strange as we think it is?", *MIT Linguistics Working Papers (MITLWP)*, 159-190.

KHANLARI, Parviz. 1995: *History of Persian language*, vol. 2, Tehran: Nashr-e-No. (In Persian).

KNESER, Reinhard, Hermann NEY, 1995: "Improved backing-of for n-gram language modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, Detroit, MI, USA, 181-184.

KNIGHT, Kevin. 1999: "Decoding complexity in word-replacement translation models", *Computational Linguistics* 25 (4), 607-615.

KOEHN, Philipp, 2010: *Statistical Machine Translation*, Cambridge University Press.

KÜBLER, Sandra, Ryan MCDONALD, & Joakim NIVRE, 2009: *Dependency Parsing*, Syntehsis Lectures on Human Language Technologies, Morgan & Claypool Publishers.

MAHAJAN, Anoop, 1995: "Toward a unified theory of scrambling" in N. CORVER & H. RIEMSDIJK (eds.): *Studies on Scrambling*, DeGruyter, 301-330.

MARTIN, Sven, Jörg LIERMANN, & Hermann NEY, 1998: "Algorithms for bigram and trigram word clustering", *Speech Communication* 24, 19-37.

MESHKATODDINI, Mahdi, 2003: *Introduction to Persian Transformational grammar*, 2nd edition, Mashhad, Iran: Ferdowsi University Press. (In Persian).

NOURIAN, Alireza, Mohammad Sadegh RASOOLI, Mohsen IMANY, & Heshaam FAILI, 2015: "On the Importance of Ezafe Construction in Persian Parsing", paper presented in *The 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7h International Joint Conference on Natural Language Processing (IJCNLP)*, Beijing, China.

NUGUES, Pierre M., 2014: *Language Processing with Perl and Prolog, Theories, Implementation, and Application*, Berlin: Springer-Verlag.

PILEVAR, Mohammad Taher, Heshaam FAILI, & Abdol Hamid PILEVAR, 2011: "TEP: Tehran English-Persian Parallel Corpus" in A. GELBUKH (ed.): *Proceedings of 12th International Conference on Intelligent Text Processing and Computational Linguistics*, Berlin/Heidelberg, 68-79.

POLLARD, Carl, & Ivan A. SAG, 1994: *Head-Driven Phrase Structure Grammar*, Chicago: University of Chicago Press.

RASOOLI, Mohammad Sadegh, Manouchehr KOUHESTANI, & Amirsaeid MOLOODI, 2014: "Persian Syntactic Treebank: A Research Based on Dependency Grammar", paper presented in *Supreme Council of Information and Communication Technology (SCICT)*, Tehran, Iran. (In Persian).

RASOOLI, Mohammad Sadegh, Manouchehr KOUHESTANI, & Amirsaeid MOLOODI, 2013: "Development of a Persian syntactic dependency Treebank" in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia: Association for Computational Linguistics, 306-314.

RASOOLI, Mohammad Sadegh, Amirsaeid MOLOODI, Manouchehr KOUHESTANI, & Behrouz MINAEI-BIDGOLI, 2011: "A Syntactic Valency Lexicon for Persian Verbs: The First Steps towards Persian Dependency Treebank", paper presented in *5th Language & Technology Conference (LTC): Human Language Technologies as a Challenge for Computer Science and Linguistics*, Poznań, Poland, 227-231.

SABEL, Joachim, & Mamoru SAITO, 2005: *The Free Word Order Phenomenon*, Berlin / New York: Mouton de Gruyter.

SAMVELIAN, Pollet, & Pegah FAGHIRI, 2013: "Introducing PersPred, a Syntactic and Semantic Database for Persian Complex Predicates" in *Proceedings of the 9th Workshop on Multiword Expressions (MWE 2013)*, Atlanta, Georgia: Association for Computational Linguistic, 11-20.

SHAMSFARD, Mehrnoush, 2011: "Challenges and Open Problems in Persian Text processing", paper presented in *5th Language & Technology Conference (LTC): Human Language Technologies as a Challenge for Computer Science and Linguistics*, Poznań, Poland, 65-69.

SCHNEIDER, Gerold, 1998: *A Linguistic Comparison of Constituency, Dependency and Link Grammar. ExtrAns Research Report: Dependency vs. Constituency*, Zürich: Institut für Informatik, Universität Zürich.

SLEATOR, Daniel, & David TEMPERLEY, 1991: *Parsing English with a link grammar*, Technical Report CMU-CS 91-196, CMU.

SOLTANZADEH, Fatemeh, 2013: *A Method to Construct Phrase Structure Treebank Using Dependency Treebank for Persian Language*. Master thesis, Sharif University of Technology, Tehran, Iran.

TABIBZADEH, Omid, 2012: *Persian Language Grammar Based on Autonomous Groups in Dependency Grammar*, Tehran: Nashr-e-Markaz. (In Perian).

VALAD, A. M., 2006: *The Syntactic Parser of Persian Language*. B.S. dissertation, Shahid Beheshti University, Tehran, Iran. (In Persian).

WINDFUHR, Gernot (ed.), 2009: *The Iranian Languages*, New York: Routledge.