# Automatic WordNet Construction Using Markov Chain Monte Carlo

Marzieh Fadaee, Hamidreza Ghader, Heshaam Faili, and Azadeh Shakery

*Abstract*—WordNet is used extensively as a major lexical resource in information retrieval tasks. However, the qualities of existing Persian WordNets are far from perfect. They are either constructed manually which limits the coverage of Persian words, or automatically which results in unsatisfactory precision. This paper presents a fully-automated approach for constructing a Persian WordNet: A Bayesian Model with Markov chain Monte Carlo (MCMC) estimation. We model the problem of constructing a Persian WordNet by estimating the probability of assigning senses (synsets) to Persian words. By applying MCMC techniques in estimating these probabilities, we integrate prior knowledge in the estimation and use the expected value of generated samples to give the final estimates. This ensures great performance improvement comparing with Maximum-Likelihood and Expectation-Maximization methods. Our acquired WordNet has a precision of 90.46% which is a considerable improvement in comparison with automatically-built WordNets in Persian.

*Index Terms*—Semantic network, WordNet, ontology, Bayesian inference, Markov chain Monte Carlo, Persian.

## I. INTRODUCTION

NOWADAYS WordNet as an ontology, where the relations between word senses are interpreted as relations between concepts, is widely used in different areas of information retrieval and linguistic researches such as machine translation, text classification, word sense disambiguation, and text retrieval.

Princeton university constructed the first WordNet in English in 1995 employing human experts [1]. In Princeton WordNet (PWN) English words have been grouped into sets of cognitive synonyms called synsets. Synsets in PWN are also interlinked by means of conceptual semantics and lexical relations. Each English word may appear in several synsets in PWN, which are realized as senses of that word.

Acknowledgment of the practicality of PWN leads many researchers to develop a WordNet in languages other than English. The obvious obstacle of developing a WordNet from scratch is that it is very labor intensive and time consuming, so different methods were proposed to construct a WordNet automatically or semi-automatically. Constructing a WordNet automatically can be categorized into two approaches: merging methods, and expanding methods. The merging methods build

the WordNet in a specific language based on monolingual resources in that language, and then map the created WordNet to existing WordNets of other languages, oftentimes PWN. The expanding methods use a basis WordNet (commonly PWN) so that they preserve the original WordNet structure, and construct the WordNet in a specific language by translating the synsets or applying different methods of learning. Resources in this category can be bilingual or multilingual. Either way having links between a WordNet in secondary language with PWN can improve the usability of said WordNet.

For example the BabelNet project [2], which uses PWN as the lexicographic resource and Wikipedia pages in different languages as the encyclopedic knowledge. It utilizes machine translation methods in order to enrich the lexical information and defines links between Wikipedia pages and WordNet items.

Although there have been several attempts in constructing a WordNet for Persian language, the lack of a sizable WordNet is still noticeable. Some of the most significant researches on Persian WordNet are introduced in [3], [4], [5].

In [3] the authors established a scoring function for ranking synsets and respective words automatically and selected the highest scores as the final WordNet. This method achieved a precision of 82.6% with manually judged candidates. In [4] an unsupervised learning approach was proposed, which constructed a WordNet automatically using Expectation-Maximization (EM) method. This research collects a set of words and their possible candidate synsets, and assigns a probability to each candidate. By applying an EM method these probabilities are updated in each iteration of the algorithm until the changes in probabilities are minute. The final WordNet was built by selecting the 10% of highly probable word-synsets and achieved a precision of 86.7%. Another project of building a Persian WordNet was FarsNet, which uses a semi-automatic method for building the Persian WordNet with some predefined heuristics and then judges each entry manually with human experts' knowledge [5].

The automatic approaches of constructing a Persian WordNet still need improvements in precision, and the manual approaches are time consuming and slow-growing. Our work aims for constructing a scalable Persian WordNet with better quality by defining a Bayesian Inference for estimating the probabilities of links between words and synsets. The proposed model is independent of language and can be applied in any language with basic resources: a raw corpus, a bilingual dictionary, and PWN.

We propose a model that utilizes a Markov chain Monte Carlo technique, namely Gibbs Sampling, in estimating the probabilities. This model iteratively disambiguates words according to their neighbors in a context and assigns probabilities to each possible sense of a word. After a certain number of iterations, the mathematical expectation of probabilities is the concluding value for each link.

In this research we construct a Probabilistic Persian WordNet, in which each Persian word is associated with relative PWN synsets and a probability that signifies these links. Using these links, the relations defined in the PWN is also applicable in our Persian WordNet.

Our proposed unsupervised approach to create a WordNet is very similar to the approach of [4] in some aspects. The main difference between these two approaches is that the Expectation-Maximization method in the research of [4] has been replaced by a fully Bayesian inference via Markov chain Monte Carlo. The Bayesian inference tries to estimate and update the probabilities assigned to word-synsets links, in an iterative process as Expectation-Maximization does. But this iterative process is a Markov chain Monte Carlo algorithm that estimates the posterior distribution. Each iteration of the algorithm includes 2 steps: (i) assigning correct senses to the words in the corpus using current estimate of the probabilities via a word sense disambiguation method, (ii) estimating new probability values according to the conditional posterior distribution, which has recently assigned senses in its condition.

Our model is expected to do better than the state-of-the-art EM method for two reasons: it incorporates the prior knowledge that the multinomial distribution over possible senses of a word is a sparse one, in its estimation of the posterior distribution, and it generates lots of samples from posterior distribution and use the expected value of these samples to give the final estimate, while the Expectation-Maximization (EM) method finds a local maximum in the search space and returns it as the final estimate. Thus, our approach takes the parameter values that may have generated the observed data with less probability into account, while the EM method fails to consider them.

Our WordNet does not have the obstacles of time and expert knowledge like the manual methods of constructing a Persian WordNet. By establishing a specified size for our WordNet (approximately 10,000 word-synset pairs) we achieve a precision of 90.46%. This is an improvement in comparison with the EM method, the state-of-the-art in constructing a Persian WordNet automatically, which achieved a precision of 86.7% with approximately the same size.

The rest of the paper is organized as follows: Section II presents an overview on several methods that have been proposed in the area of automatic and semi-automatic WordNet construction. Section III presents the details of the proposed model for automatically constructing the WordNet, and the training algorithm. Section IV explores experimental results and evaluations. Lastly, the work is concluded and some future works are suggested.

## II. BACKGROUND

WordNet is a semantic network providing machine-readable lexicographic information, first developed in Princeton University [1] . Princeton WordNet is a lexical database consisting of syntactic categories of English words—nouns, verbs, adjectives and adverbs, grouped into lexicalized concepts. Each cognitive synonym (synset) conveys a conceptual meaning and is part of a relational network. In WordNet several hierarchal relations are defined between synsets and words, such as synonymy (similar), antonymy (opposite), hyponymy (subordinate) and meronymy (part).

Princeton WordNet is currently the most advanced English WordNet containing a wide range of English words and word senses. It was created manually by English linguists, but manual construction of WordNet is a time consuming task and requires linguistic knowledge. In order to achieve comprehensive WordNet in languages other than English, many automatic and semi-automatic approaches were presented.

EuroWordNet was a similar project but with the goal of enriching the resources of Western European languages [6]. It started with four languages: Dutch (at the University of Amsterdam), Italian (CNR, Pisa), Spanish (Fundacion Universidad Empresa), and English (University of Sheffield, adapting the Princeton WordNet); and later Czech, Estonian, German, and French were added.

By applying a common framework between all WordNets and integrating them into a single database, EuroWordNet became a multilingual semantic network which could be used in many multilingual applications.

In order to maintain a similar coverage in all languages, first a set of 1,024 base concepts were created. Since these concepts were not lexicalized in all languages, iteratively, the base concepts were selected based on the common concepts between the majority of European languages. The base concepts were classified with the aid of a language-independent top ontology.

EuroWordNet is not used widely due to licensing issues and lack of further extensions. In [7] a freely-available French WordNet (WOLF) was built automatically from multilingual resources like Wikipedia and thesaurus. In the proposed approach, they constructed a multilingual lexicon by aligning a parallel corpus for five languages. By using multiple languages, polysemous lexicon entries are disambiguated. WOLF contains all four parts of speeches, including 32,351 synsets corresponding with 38,001 unique literals. The average polysemy in WOLF is 1.21 synsets per literal but the core vocabulary of it is sparse.

The resulting WordNet was evaluated both automatically and manually. In the former approach, they compared WOLF with the French WordNet, created as part of the EuroWordNet project, in regard to the words that appeared in WOLF so as

not to penalize the WOLF for not containing some words in the utilized corpora. WOLF achieved a precision of 77.1% and recall of 70.3%. In the latter approach they randomly selected 100 literals and the corresponding 183 synsets to judge them by hand and achieved 80% correctness in assigned synsets. In this paper it is shown that building a WordNet with the alignment approach provides more basic synsets.

BalkaNet was another European project focusing on Central and Eastern European languages [8]. The method of constructing BalkaNet is comparable to EuroWordNet with added features such as independence of every WordNets. It uses individual monolingual WordNets that have already been developed for the participant languages, including Greek, Turkish, Romanian, Bulgarian, Czech and Serbian. BalkaNet contains 15,000 comparable synsets in each language, and 30,000 literals. BalkaNet concept sets are very dense in the sense that for any concept in the BalkaNet concept sets, all of its hypernyms are also in the BalkaNet. Turkish WordNet is a side-result of the BalkaNet project [9] containing 11,628 synsets and 16,095 literals. It has an average polysemy of 1.38.

Word sense disambiguation techniques are applied in many approaches of constructing or expanding a WordNet. In [10] they defined multiple heuristics including maximum similarity, prior probability, sense ordering, IS-A relation, and co-occurrence, for linking Korean words to English synsets. The heuristics were then combined using decision tree learning for non-linear relationship. To evaluate each of their proposed heuristics separately, they manually judged the candidate synsets of 3260 senses. The decision tree based combination of the heuristics achieved 93.59% in precision and 77.12% in recall. Their generated Korean WordNet contains 21,654 synsets and 17,696 nouns.

There are other attempts in constructing WordNets for Asian languages. A Thai WordNet was constructed utilizing machine-readable dictionaries [11]. In this semi-automatic approach several criteria were defined to extract and evaluate relations between translated words. To evaluate the candidate links in each criterion they apply the stratified sampling technique [12]. The final WordNet contains 19,582 synsets and the corresponding 13,730 words and provides 80% coverage and 76% accuracy.

Arabic WordNet was first introduced in [13]. By considering three main criteria—connectivity, relevance and generality, synsets were extracted and manually validated. In this project they also generated a machine-understandable semantics in first order logic for word meanings. The Arabic WordNet consists of 11,270 synsets and 23,496 Arabic expressions. There were several extensions of Arabic WordNet, particularly the semi-automatic approach in [14]. They designed a Bayesian network with four layers to equate Arabic words and English synsets by using lexical and morphological rules. The resulting WordNet has a precision of 67%.

There were several researches on constructing WordNet in Persian language in recent years, focusing on Persian adjectives [15], verbs [16], or nouns [17]. These methods either use lexicographers' knowledge in constructing the WordNet manually, or proposing semi-automatic approaches. PersiaNet was a project of Princeton University for a comparable Persian WordNet with Princeton WordNet [17]. This work, which is strictly based on a volunteering participation of experts, has a scarce lexical coverage. It uses Persian orthography for representing words and also supports a parallel Roman writing system in order to facilitate searching for Persian words.

In [18] a semi automatic method was proposed using human annotators to make the decision of relativeness of each word and candidate synsets. In this work they introduced FarsNet which consists of two parts: semantic lexicon and lexical ontology. They used a bilingual dictionary, a syntactic lexicon including the POS tags of the entries, a Persian POS tagged corpus and WordNet in order to develop an initial lexicon and perform word-sense disambiguation. A linguistic expert reviewed the results to evaluate the method which gained a 70% accuracy.

They expanded their work later, completing FarsNet by applying some heuristics and word sense disambiguation techniques in an automated method with human supervision [5]. It consists of 9,266 synsets and 13,155 words. In this paper we use FarsNet as the baseline in evaluating the quality of our WordNet.

In [3] an automatic method was presented in which they compute a similarity score between each Persian word and the candidate English synsets and select the highest score as the respective link. The score, containing the mutual information of words, is computed from bilingual dictionaries and Persian and English corpora. To evaluate the constructed Persian WordNet they randomly selected 500 Persian words and assessed the accuracy of the selected synsets. The precision of unambiguous links between words and synsets is 95.8%, and of ambiguous links is 76.4%. In total they achieved an accuracy of 82.6%.

In [4] an unsupervised learning approach was proposed for constructing a Persian WordNet. In this work they first assemble a list of candidate English synsets for each Persian word using a bilingual dictionary and Princeton WordNet. In the next step they automatically connect English synsets with Persian words using Expectation-Maximization method and eliminates unrelated links. The probabilities of each link are calculated in the Expectation step from the information extracted from a Persian corpus. In the Maximization step, the probabilities of selected candidate synsets is updated. In order to evaluate the resulting WordNet they manually judged 1365 randomly selected links between words and synsets. By accepting the top 10% of the probable links as the final Persian WordNet 7,109 literals (from 11,076 words appeared in the corpus) and 9,427 synsets were selected. The WordNet accuracy is 89.7% for adjectives, 65.6% for adverbs and 61.2% for nouns. This approach strongly depends on the initial Persian corpus that is used in the Expectation step and the initial values of probabilities of links.

## A. Markov chain Monte Carlo

Using Bayesian inference to estimate posterior over a model, one may come across a situation in which the posterior or an intermediate distribution could not be computed analytically. In these cases, a widely used method is to estimate the intended distribution using Markov chain Monte Carlo techniques. The works [19], [20] are examples of dealing with this kind of situation in Bayesian inference. In [20] two MCMC techniques are used to induce a probabilistic context free grammar in an unsupervised setting. In this work, the MCMC methods are employed to sample from posterior distribution over probabilities of CFG rules and sequence of parse trees conditioned on the observed data. In [19] an MCMC technique is put in action to find a MAP estimation of a posterior distribution over POS tag sequence conditioned on the observed data.

In order to estimate a distribution using MCMC techniques, the techniques are used to generate sufficient samples from the distribution. The MCMC techniques construct a Markov chain whose desired distribution is equal to the distribution intended to be sampled. This means that they provide the transition probability between states of Markov chain so that the probability of visiting a state $St_i$ of the chain be $p(St_i)$, according to the desired distribution. Then, they generate samples by moving between states of the Markov chain. Of course, some runout steps are required for the model to take, before that the generated samples being from the stationary distribution of the Markov chain. After generating sufficient samples from the desired distribution, these probabilistic choices can be used to calculate expectation over states. This makes the method a Monte Carlo technique. For example in [20], the sample values for $\theta$, a random variable corresponding to the probability of CFG rules, are used to compute the expectation over it. Then, the resulted expectation is used as the estimated value for probability of CFG rules.

*1) Gibbs Sampling:* Gibbs sampling [21] is a sampling technique from class of Markov chain Monte Carlo techniques. This technique is used in situations that the state of the model is comprised of multiple random variables [22]. In other words, the situations in which the joint distribution of multiple random variables is intended to be sampled. If we assume that each state in the model has k dimension or is a joint of k random variables, the basic idea in this technique is to sample each random variable involved in the state of the model separately, but conditioned on the other $k-1$ dimensions [22]. That is to sample each random variable from the following distribution:

$$P(rv_i|rv_1^t, ..., rv_{i-1}^t, rv_{i+1}^{t-1}, ..., rv_k^{t-1}).$$

Here the superscript corresponds to time. It also provides the information that how many samples are generated from a random variable until current time. After sampling each random variable once, using the conditional distribution above, we will have one sample from the joint distribution of random variables. Repeating this action for a sufficient number of times, we will generate sufficient samples from the join distribution. These samples can be used in a variety of ways to compute an estimation of the intended distribution.

## B. Dirichlet Priors

In recent years, the Bayesian methods for estimating probabilities are widely favored over the maximum likelihood estimation method among scientists working in computational linguistics [19], [23]. One reason for this, is the fact that Bayesian methods provide a way to take the prior knowledge about the model into account when doing estimation. As a standard example, taken from [23], suppose that we are given a coin to decide whether it is fair or not. Tossing the coin 10 times, we observe this sequence of heads and tails: (T T T T T T H H H H). Maximum likelihood estimation gives an estimate of 0.6 for the probability of observing tail in next toss. Maximum likelihood results this estimate by calculating the parameter value that is most likely to generate observed data. If we take $\theta$ as the probability of observing tail, that means

$$\hat{\theta} = \arg\max_{\theta} P(D|\theta).$$

As one can see, no prior knowledge is incorporated in this estimation. This is while the Bayesian methods take the prior knowledge into account using Bayes rule. For example maximize a posteriori estimation gives an estimate of $\theta$ as follows:

$$
\begin{aligned}
\hat{\theta} &= \arg\max_{\theta} P(\theta|D) \\
&= \arg\max_{\theta} \frac{P(D|\theta)P(\theta)}{P(D)} \\
&= \arg\max_{\theta} P(D|\theta)P(\theta).
\end{aligned}
$$

This estimation provides the possibility that our expectation of what $\theta$ could be, affect the final estimated value for $\theta$. Here, by using a Beta distribution, which is a two dimensional version of dirichlet distribution, we can put a prior expectation toward fairness or unfairness of the coin into the estimation. If we choose parameters of the Beta distribution to be near zero, this will put a prior in favor of unfairness of the coin into the estimation. This means that an estimate of $\theta$ nearer to 0 or 1 is more desirable. This characteristic of Bayesian methods makes them more appropriate than maximum likelihood estimation, because it provides the possibility of using linguistically appropriate priors.

## III. MODELING THE PROBLEM OF AUTOMATICALLY CONSTRUCTING A WORDNET

### A. The Approach

In this work, we create a probabilistic WordNet in which each link between a word and its synsets has a probability assigned to it. This probability signifies the relatedness of each synset to the word. The proposed approach consists of the following steps:

1) Collect candidate synsets as possible senses for each Persian word
2) Compute the probability of relatedness of each synset to a particular word (iteratively)
3) Choose the highest word-synset probabilities as the final WordNet

For the first step, we use a bilingual dictionary to find all possible definitions of a Persian word in English. Then, we use the Princeton WordNet to find all senses of the English words and consider them as potential senses of the Persian word.

In the next step, we compute the probabilities of the Persian word having each of the senses extracted in the previous step. These probabilities are computed according to different senses of a word that appear in a raw corpus. This corpus contains Persian words and the POS tags for each word, and so it aids in building a POS-aware Persian WordNet.

We use a word sense disambiguation technique, previously utilized in [4], as part of a Bayesian model in an unsupervised configuration to assign the correct sense of each word based on the context in which the word appears. During the process of sense disambiguation of words in the corpus, we compute the probabilities of assigning different senses to a word. As a result, some senses of a word will be assigned very small probabilities in comparison with other senses.

The algorithm iteratively computes the new probabilities using Gibbs Sampling, which will be discussed in the following section.

The algorithm uses a Markov Chain Monte Carlo (MCMC) method, Gibbs sampling, and iteratively computes the new probabilities. MCMC methods are widely used to estimate probability distributions that could not be computed analytically.

In the final step, we eliminate the senses assigned small probabilities according to some measures, and use the remaining senses to build up the Persian WordNet.

### B. The Model

We define the probabilities of assigning synsets $s_1, ..., s_n$ to a word $w$ as

$$\theta_w : [\theta_{ws_1}, \theta_{ws_2}, ... \theta_{ws_n}]. \tag{1}$$

If $t_w$ indicates a synset assigned to word w, $t_w|context$ will have a multinomial distribution whose parameters are in the vector $\theta_w$. For ease of reference, we present our notation in Table 1. For a multinomial with parameters $\theta_w = [\theta_{ws_1}, ..., \theta_{ws_k}]$ a natural choice of prior is the K-dimensional Dirichlet distribution, which is conjugate to the multinomial [19]. If we assume that the dirichlet distribution is symmetric and its K parameters are equal to $\alpha$, $\alpha < 1$ will favor sparse multinomial distributions for $\theta_w$. As the distribution of senses of a word in a context is a sparse multinomial distribution, a Dirichlet distribution with $\alpha < 1$ will be a linguistically appropriate prior in our model.

So we can safely assume that $\theta_w$ has a $Dirichlet(\alpha_w)$ distribution:

$$\theta_w \sim Dirichlet(\alpha_w). \tag{2}$$

Suppose that $\theta$ is the vector of $\theta_w$ for all words. The goal of constructing the WordNet in our model is obtaining a wise $\theta$ which can be computed as follows:

$$P(\theta|W) = \sum_t P(t, \theta|W), \tag{3}$$

with $W$ being the corpus we use for sense disambiguation and $t$ is a tag sequence of synsets. The distribution on the right side of the equation could be written as follows:

$$\begin{aligned} P(t, \theta|W) &= \frac{P(W|\theta, t)P(\theta, t)}{P(W)} \\ &= \frac{P(W|\theta, t)P(\theta, t)}{\sum_{t,\theta} P(W|\theta, t)P(\theta, t)}, \end{aligned} \tag{4}$$

which is intractable because of the large possible sets of $t$ and $\theta$, which should be observed to compute the denominator. If we take the Dirichlet prior into account, the posterior distribution will change as follows:

$$P(t, \theta|W, \alpha), \tag{5}$$

where $\alpha$ is a vector of parameters of the prior distributions which are Dirichlet distributions.

Since computing the probability distribution of Equation 5 is intractable, we propose to use a Markov chain Monte Carlo algorithm, Gibbs sampling, to generate samples from this distribution and use the samples to compute a wise value for $\theta$. To use the Gibbs sampling method to generate samples from $P(\theta, t|W, \alpha)$, we have to sample each random variable conditioned on the current value of the other random variables constituting the state of the model. This means that we have to sample from the following two probability distributions at each step:

$$P(t|\theta, W, \alpha), \tag{6}$$
$$P(\theta|t, W, \alpha). \tag{7}$$

Formula (6) illustrates the sense assignments' probabilities, and Formula (7) illustrates the candidate senses' probabilities. In the following section the sampling process of these two distributions and how we estimate the probabilities are discussed in detail.

### C. Estimating the Probabilities

In order to generate samples from (7) we can independently sample each multinomial distribution $P(t_i|w_i, context_{w_i}, \theta)$ for all possible $i$. Then we use resulted value for each $t_i$ as sense tag of $w_i$. In the next step, given a value for $t$ we generate sample from (8).

TABLE I
NOTATION

| | |
|---|---|
| $w$ | Persian word. |
| $wordlist$ | the list of all Persian words we want to include in our WordNet. |
| $s_i$ | Princeton WordNet synset. |
| $ws_i$ | assigning synset $s_i$ to Persian word $w$ as a possible sense of that word. |
| $\theta_{ws_i}$ | probability of relateness of $ws_i$ |
| $\theta_w$ | vector of probabilities of candidate senses for word $w$ |
| $\theta$ | vector of $\theta_w$s for all words. |
| $\alpha_w$ | Dirichlet parameter for the distribution of $\theta_w$ |
| $\alpha$ | vector of $\alpha_w$s for all words $w$ in $wordlist$ |
| $W$ | corpus, providing words $w$ for disambiguation. |
| $t_w$ | a random variable whose possible values are candidate senses $s_i$ of word $w$ |
| $t$ | vector of $t_w$s for all words $w$ in $wordlist$ |

*1) Sense Assignment Probability Estimation:* In the literature, a key assumption to induce correct sense of a word is that the context surrounding the word is indicative of its correct sense [23]. Making the same assumption, the distribution of senses of a word conditioned on the word itself and its context will be independent from the other words of the corpus and their senses. So we can write:

$$P(t|\theta, W, \alpha) = \prod_i P(t_{w_i}|w_i, context_{w_i}, \theta). \quad (8)$$

Hence we involve the context in computing the probabilities by considering a window of words rather than every individual word.

Finding multinomial distribution $P(t_{w_i}|w_i, context_{w_i}, \theta)$ for each possible $i$, and using the distributions to interpret the correct sense of each word $w_i$ could be viewed as a word sense disambiguation task.

Word sense disambiguation is the task of interpreting senses of a word in a context via supervised or unsupervised methods. Most words in the Persian language are polysemous, so, in order to differentiate between individual meanings of a word we need to consider disambiguating its senses. To attain this goal we use an ancillary Persian corpus, Bijankhan [24], as our corpus for extracting statistical information. For every word in the training set, windows of words are obtained from the corpus containing the neighbors of that particular word for every occurrence of it in the corpus.

The score of each word $w$ and synset $s$ is calculated from the following formula:

$$score(w, s) = \frac{\sum_{w'}\sum_v \theta_{w',s} \times PMI(w', v)}{n}, \quad (9)$$

where $w'$ is a word that has $s$ as its candidate synset, $n$ is the number of these words, $v$ is a word in the window of $w$, *PMI* is point-wise mutual information, and $\theta_{w',s}$ is the probability assigned to word $w'$ and sense $s$ in the previous iteration of the procedure. This score function disambiguates a word by considering the senses of the neighbors of the word in a context [4].

Using the scores computed in Equation 9, we can find the distributions

$$P(t_{w_i}|w_i, context_{w_i}, \theta)$$

for all possible $i$ by means of the following formula:

$$P(t_{w_i}|w_i, context_{w_i}, \theta) = \frac{score(w_i, t_{w_i})}{\sum_j score(w_i, s_j)}. \quad (10)$$

By involving all contexts in which a word is used in a corpus—windows of words—individual senses of the word have the chance of obtaining acceptable probabilities in the computation. For better determining individual senses of every word we consider the parts of speech of them. One of the main attributes of every synsets in Princeton WordNet is the part of speech of that sense. To take heed of this attribute we consider individual parts of speech for every word and perform the sampler on words and synsets in regard to parts of speech.

*2) Candidate Sense Probability Estimation:* In order to compute the second distribution we assume that $\theta_w$ for each word is independent from the others and as we discussed prior distribution on $\theta_w$ is Dirichlet distribution. So the prior probability on $\theta$ could be written as follows:

$$\begin{aligned} P(\theta) &= \prod_{w \in wordList} P(\theta_w|\alpha_w) \\ &= \prod_{w \in wordList} \left( \prod_{s \in senses_w} \frac{1}{B(\alpha_w)} \theta_s^{\alpha_s - 1} \right), \quad (11) \end{aligned}$$

where

$$B(\alpha_w) = \frac{\prod_{s \in senses_w} \Gamma(\alpha_s)}{\Gamma(\sum_{s \in senses_w} \alpha_s)}, \quad (12)$$

*wordList* is the list of all words we want to include in our WordNet. As a result to formulation above, $P(\theta)$ will be Dirichlet distribution and could be written as $P_D(\theta|\alpha)$, where $\alpha$ is a vector containing $\alpha_w$ for all words in $wordList$ and is the parameter of the prior Dirichlet distributions of $\theta_w$.

Since the prior distribution of $\theta$ is conjugate prior to the likelihood of the sense tags, the posterior on $\theta$ conditioned on sense tags will be a Dirichlet distribution:

$$P(\theta|t, W, \alpha) \propto P(t, W|\theta, \alpha)P(\theta|\alpha); \quad (13)$$

$$\begin{aligned} &\propto \left( \prod_{w \in wordList, s \in senses_w} \theta_s^{Cn_{w \to s}(t)} \theta_s^{\alpha_s - 1} \right) \\ &= \prod_{w \in wordList, s \in senses_w} \theta_s^{Cn_{w \to s}(t) + \alpha_s - 1}, \end{aligned}$$

which could be written as

$$
\begin{aligned}
P(\theta|t, W, \alpha) &= P_D(\theta|Cn(t) + \alpha) \\
&= \prod_{w \in wordList} P_D(\theta_w|Cn_w(t) + \alpha_w), \quad (14)
\end{aligned}
$$

where $Cn_w(t)$ is a vector of the number of times the assignment $w \rightarrow s_i$, where $s_i \in senses_w$, happened in the context and $Cn(t)$ is a vector of $Cn_w(t)$ for all $w$.

For this to be done, we sample each Dirichlet distribution, $P_D(\theta_w|Cn_w(t) + \alpha_w)$, independently and put the results together to constitute a sample from $\theta$. To create sample from the Dirichlet distribution, we use a Gamma distribution and sample $\gamma_{s_j}$ from $Gamma(\alpha_{s_j} + Cn_{w \rightarrow s_j}(t))$ for all $\alpha_{s_j} \in \alpha_w = (\alpha_{s_1}, ..., \alpha_{s_m})$ and finally set each $\theta_{s_j} \in \theta_w = (\theta_{s_1}, ..., \theta_{s_m})$ as follows:

$$
\theta_{s_j} = \frac{\gamma_{s_j}}{\sum_{i=1}^{m} \gamma_{s_i}}. \quad (15)
$$

So we use Formulas (8), (10), and (14) to generate samples from Formulas (6) and (7). This will result in samples from $P(\theta, t|W, \alpha)$ as was discussed before. After sampling the acceptable number of values for $\theta$, we can compute the Expectation of $\theta$ over these values which would grant us the wise $\theta$ we were looking for.

## IV. EXPERIMENTS AND EVALUATION

In this paper we have conducted different experiments to evaluate our proposed method. This section carries out with introducing the environment of the experiments and details on different trials and the methods of evaluation applied in this project.

### A. Train Dataset

We use Bijankhan dataset [24] to take into account the context for every word in order to perform an unsupervised word sense disambiguation and compute $\theta$ values in the iterative process. Bijankhan is a POS tagged corpus in the Persian language consisting of news and colloquial texts. It contains 500 documents and around 2.6 million manually tagged words. The tag set consists of 40 different Persian POS tags, however we only use the four main POS tags in this experiment; verb, noun, adjective, and adverb.

### B. Test Dataset

To evaluate the accuracy of our WordNet we use a manual approach of assessing our results, in view of the fact that if we wanted to automatically evaluate the WordNet we had to compare the results with the existing Persian WordNets, which wasn't fair to our WordNet; by comparing our results with the previous WordNets we would penalize the correctly assigned word-synset pairs that do not exist in the earlier WordNets.

To avoid this, we opt for building a test set which we have based on FarsNet, the semi automatically constructed WordNet. FarsNet links are added to this test set as the *correct*

TABLE II
STATISTICS OF THE TEST DATA SET

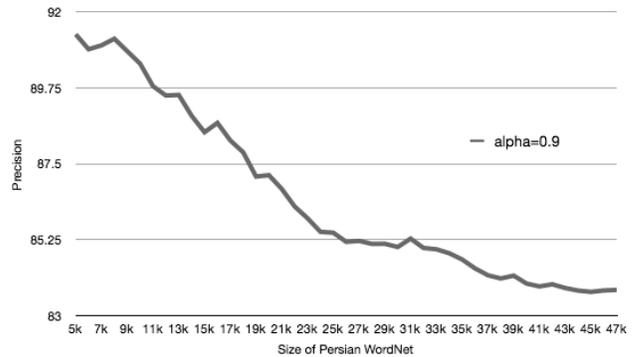| | |
|---|---|
| Number of incorrect links (0) | 3482 |
| Number of correct links (1) | 17393 |
| size of the test dataset | 20874 |



Fig. 1. Precision of Persian WordNet with respect to $N$, the size of WordNet (the $N$ most probable word-synset links) after 100 iterations

*links*. We also judged a subset of assigned words - synsets links manually and added this information to the test set. Our final gold data contains 3482 incorrect links and 17393 correct links.

### C. Results

Upon the termination of the algorithm, a WordNet in target language and the probabilities of assigning each candidate synsets to each word are acquired and are sorted based on the probabilities, so by selecting the $top - N$ most probable word-synset pairs we obtain our Persian WordNet. The parameter $N$ determines the size of the WordNet; there is a trade-off between precision of the WordNet and the coverage over all Persian Words i.e. the size of the WordNet, $N$.

We define the precision of the WordNet as the number of assigned links in the WordNet which appeared in the test data as correct links divided by the total number of assigned links in the WordNet which appeared in the test data. This definition of precision for WordNet was also used in BabelNet project [2].

Figure 1 demonstrates the precision of our WordNet with respect to size of the WordNet. We can observe that by increasing the size of the WordNet, precision decreases which is expected. By selecting the first 10,000 word-synset pairs we have a WordNet of precision 90.46%. This is an improvement in comparison with the state-of-the-art automatically built Persian WordNet which gained precision of 86.7% with approximately the same size of the WordNet [4].

### D. Dirichlet Prior Parameter Tuning

In this section we evaluate the effect of the Dirichlet parameter in our proposed model. As we have stated earlier, dirichlet prior is taken into service to provide the possibility
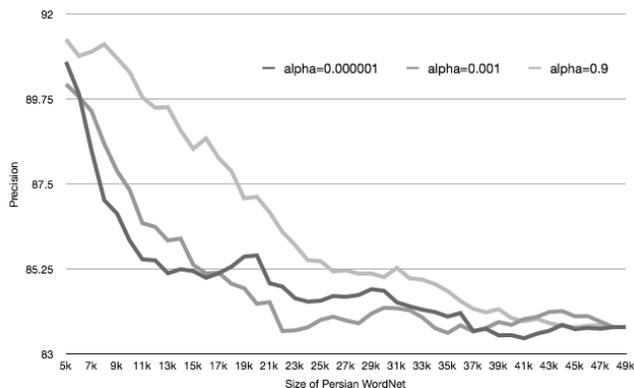
19

Marzieh Fadaee, Hamidreza Ghader, Heshaam Faili, and Azadeh Shakery



Fig. 2. Precision of Persian WordNet with respect to $N$, the size of WordNet after 100 iterations
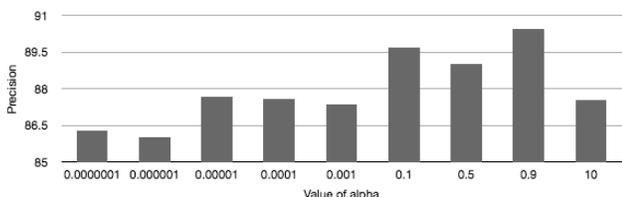


Fig. 3. Precision of Persian WordNet of size 10,000 with different values of $\alpha$



Fig. 4. Comparison of precision of Persian WordNet with respect to $N$ for different number of iterations
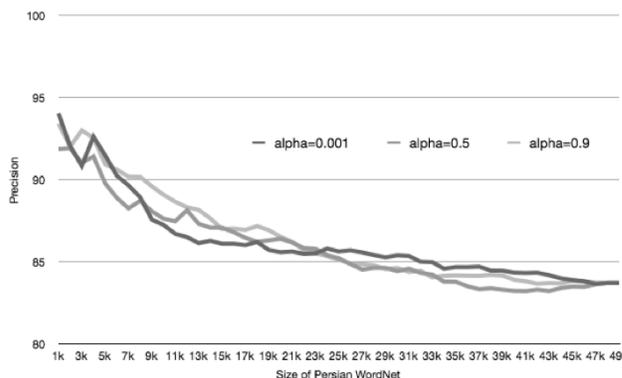


Fig. 5. Precision of Persian WordNet with respect to $N$, the size of WordNet after 500 iterations

of incorporating linguistically appropriate priors. According to the discussion, a valid assumption about the distribution of senses of a word is to assume that it is a sparse multinomial distribution.

Dirichlet distribution with parameters smaller than 1 is a natural prior over parameters of a sparse multinomial distribution. So, we assume a K-dimensional Dirichlet distribution over parameters of multinomial distribution with K dimensions. For simplicity, we assume that all dirichlet distributions are symmetric and its parameters are equal to $\alpha$. As we prefer sparse multinomial distributions over senses of a word, we set $\alpha < 1$ for all Dirichlet prior distributions, but we also experiment with some large $\alpha$s to observe the differences.

In order to observe the effect of the Dirichlet parameter, Figure 3 presents different values of precision of the WordNet with a fixed size of 10,000 word-sense pairs for different values of $\alpha$. We can observe that the precision of the WordNet increases with the increase of the Dirichlet parameter. With optimum value of $\alpha$, we obtained a precision of 90.46%.

The precision of the automatically built WordNet in this paper is calculated based on the joined test set containing annotated gold data, FarsNet, and the set of randomly judged words by human experts. $N$ top demonstrates the size of the WordNet, for instance at $N = 10000$ we are selecting the 10,000 top links of word-sense and regarding them as our WordNet. It is clear that by expanding the size of the WordNet
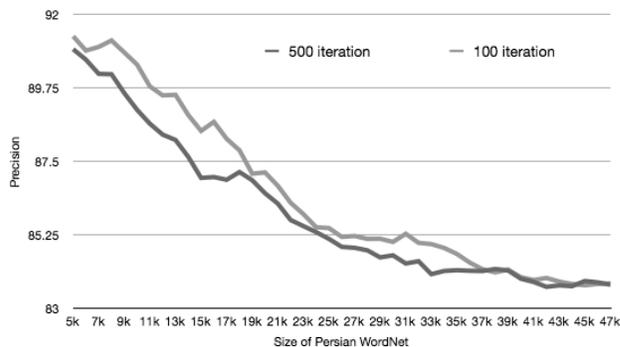
and introducing more senses into our selected links we lose precision.

### E. Number of Iterations

As stated earlier, the number of iterations of the proposed algorithm has an effect on the final results. In order to observe the effect of number of iterations on the results, we choose the approximate optimum value of 0.9 for $\alpha$ and present Figure 4. It is clear from this figure that the higher number of iterations acquire roughly the same results as lower number of iterations. The probabilities of the word-sense links are already converged with 100 iterations and we can trust our results with 100 iterations.

This figure shows that even with higher number of iterations we achieve better precision in the first 1000 links, but the value of precision gradually decreases with respect to lower number of iterations, hence, with 100 iterations of the algorithm we achieve better precision after the first 4000 links.

### F. Coverage of the WordNet

To evaluate the coverage of our WordNet over all Persian words we perform two types of assessments: Considering

all words appearing in a Persian corpus, Bijankhan, as the baseline for Persian words, and analyzing the coverage of our WordNet over FarsNet as the baseline.
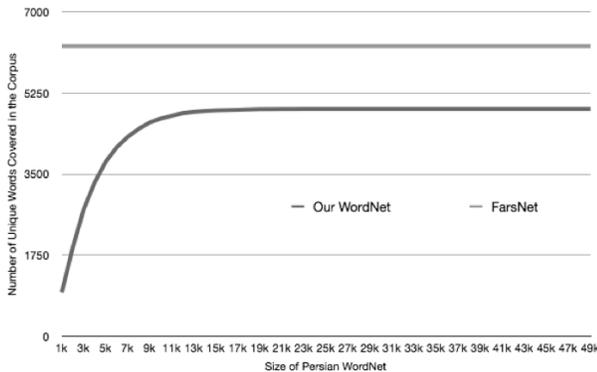


Fig. 6. Coverage of FarsNet and our Persian WordNet with respect to $N$, the size of WordNet, for $\alpha = 0.9$ over Persian words that appear in Bijankhan corpus

Figure 6 shows the number of unique words of the corpus, covered by our WordNet and also covered by FarsNet. We can observe that with high precision at the size of 10,000, our method covers a little less than FarsNet, which is a semi-automatically built WordNet.
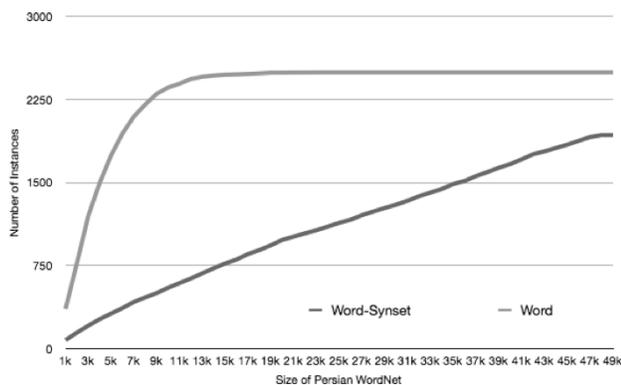


Fig. 7. Coverage of our Persian WordNet with respect to $N$ the size of WordNet, for $\alpha = 0.9$ over FarsNet

The coverage of our WordNet in comparison with FarsNet as the baseline is displayed in Figure 7. In this figure we perceive two types of coverage: word, and word-synset pair. The former testifies to the number of words that both our WordNet and FarsNet have in common, and the latter testifies to the common sense coverage between two WordNets.

Figure 7 illustrates this experiment. We can note that by selecting 10,000 links, we cover 2,357 unique words in FarsNet, and this value only increases slightly by the increase of the size of our WordNet. However, the number of word-sense pairs covered in both FarsNet and our WordNet gradually increases with the increase of the size of our

WordNet, signifying that we are adding new senses to the existing words with increase of the size and including new links.

## V. CONCLUSION

In this paper, we have presented a method for constructing a Persian WordNet automatically. This method, which is based on a Bayesian Inference, uses Gibbs Sampling as a Markov chain Monte Carlo technique in order to estimate the probabilities of senses for each word in Persian. The final WordNet is established by selecting the pairs of word-synsets with highest probabilities. Our experiments show that this WordNet has satisfactory coverage over Persian words and maintains higher precision in comparison with published automatically-built WordNets in Persian. The resulting WordNet is freely released and can be downloaded from our website.[1]

In this paper, we assumed sparse multinomial distributions over senses of all words and used the same value for the parameters of all Dirichlet priors. In reality, the degree of sparsity of multinomial distributions differs for different words, and we should take this into account when setting parameter values of Dirichlet distributions as priors.

Another proposal for future work is to use variational Bayes as inference method for training the model. This will mitigate the problem of slow convergence of training step, which is the result of using Gibbs sampling as the inference algorithm. This makes the model capable of learning semantic nets with larger amount of words in relatively shorter time.

## REFERENCES

[1] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, pp. 39–41, November 1995. [Online]. Available: http://doi.acm.org/10.1145/219717.219748

[2] R. Navigli and S. P. Ponzetto, "BabelNet: Building a very large multilingual semantic network," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 216–225.

[3] M. Montazery and H. Faili, "Automatic Persian WordNet construction," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, ser. COLING '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 846–850. [Online]. Available: http://dl.acm.org/citation.cfm?id=1944566.1944663

[4] ——, "Unsupervised learning for Persian WordNet construction," in *RANLP*, G. Angelova, K. Bontcheva, R. Mitkov, and N. Nicolov, Eds. RANLP 2011 Organising Committee, 2011, pp. 302–308.

[5] M. Shamsfard, A. Hesabi, H. Fadaei, N. Mansoory, A. Famian, S. Bagherbeigi, E. Fekri, M. Monshizadeh, and M. Assi, "Semi automatic development of FarsNet, the Persian WordNet," in *5th Global WordNet Conference (GWA2010)*, Mumbai, India, 2010.

[6] P. Vossen, Ed., *EuroWordNet: A multilingual database with lexical semantic networks*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[1] *http://ece.ut.ac.ir/nlp*

Marzieh Fadaee, Hamidreza Ghader, Heshaam Faili, and Azadeh Shakery

[7] B. Sagot and D. Fišer, "Building a free French WordNet from multilingual resources," in *OntoLex 2008*, Marrackech, Morocco, 2008.

[8] S. Stamou, K. Oflazer, K. Pala, D. Christoudoulakis, D. Cristea, D. Tufis, S. Koeva, G. Totkov, D. Dutoit, and M. Grigoriadou, "Balkanet: A multilingual semantic network for the balkan languages," in *Proceedings of the 1st Global WordNet Association conference*, 2002.

[9] O. Bilgin, Ö. Ç. Glu, and K. Oflazer, "Building a Wordnet for Turkish," pp. 163–172, 2004.

[10] C. Lee, G. Lee, S. JungYun, and G. Leer, "Automatic WordNet mapping using word sense disambiguation," in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*, 2000.

[11] P. Sathapornrungkij, "Construction of Thai WordNet Lexical Database from Machine Readable Dictionaries," *English*, pp. 87–92, 2005.

[12] R. V. Krejcie and D. W. Morgan, "Determining sample size for research activities," *Educational and Psychological Measurement*, vol. 30, no. 3, pp. 607–610, 1970. [Online]. Available: http://eric.ed.gov/ERICWebPortal/recordDetail?accno=EJ026025

[13] W. Black, S. Elkateb, A. Pease, H. Rodriguez, and M. Alkhalifa, "Introducing the Arabic WordNet project," *Word Journal Of The International Linguistic Association*, 1998.

[14] H. Rodriguez, D. Farwell, J. Farreres, M. Bertran, M. Alkhalifa, and A. Marti, *Arabic WordNet: Semi-automatic Extensions using Bayesian Inference*. European Language Resources Association (ELRA), 2008, pp. 1–3. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2008/

[15] A. Famian, "Towards Building a WordNet for Persian Adjectives," *International Journal of Lexicography*, no. 2000, pp. 307–308, 2006.

[16] M. Rouhizadeh, M. Shamsfard, and M. Yarmohammadi, "Building a WordNet for Persian verbs," in *the Proceedings of the Fourth Global WordNet Conference (GWC '08)*. The Fourth Global WordNet Conference, 2008, pp. 406–412.

[17] F. Keyvan, H. Borjian, M. Kasheff, and C. Fellbaum, "Developing PersiaNet: The Persian WordNet," in *3rd Global wordnet conference*. Citeseer, 2007, pp. 315–318.

[Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.148.7473\&amp;rep=rep1\&amp;type=pdf

[18] M. Shamsfard, "Towards semi automatic construction of a lexical ontology for Persian," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco: European Language Resources Association (ELRA), may 2008, http://www.lrec-conf.org/proceedings/lrec2008/.

[19] S. Goldwater and T. Griffiths, "A fully Bayesian approach to unsupervised part-of-speech tagging," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 744–751.

[20] M. Johnson, T. Griffiths, and S. Goldwater, "Bayesian inference for PCFGs via Markov chain Monte Carlo," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, New York: Association for Computational Linguistics, April 2007, pp. 139–146. [Online]. Available: http://www.aclweb.org/anthology-new/N/N07/N07-1018.bib

[21] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," in *Readings in computer vision: issues, problems, principles, and paradigms*, M. A. Fischler and O. Firschein, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987, pp. 564–584. [Online]. Available: http://dl.acm.org/citation.cfm?id=33517.33564

[22] P. Resnik and E. Hardisty, "Gibbs sampling for the uninitiated," University of Maryland, Tech. Rep., Oct. 2009.

[23] S. Brody and M. Lapata, "Bayesian word sense induction," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, ser. EACL '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 103–111. [Online]. Available: http://dl.acm.org/citation.cfm?id=1609067.1609078

[24] M. BijanKhan, "The role of the corpus in writing a grammar: An introduction to a software," *Iranian Journal of Linguistics*, vol. 19, 2004.