MirasText: An Automatically Generated Text Corpus for Persian

Behnam Sabeti, Hossein Abedi Firouzjaee, Ali Janalizadeh Choobbasti,
S.H.E. Mortazavi Najafabadi, Amir Vaheb
Miras Technologies International
NO. 3, 2nd Alley, North Sheikh Bahai St., Tehran, Iran
{behnam, hosein, ali, hani, amir}@miras-tech.com

Abstract

Natural Language Processing is one of the most important fields of artificial intelligence. The rapid growth of digital content has made this field both practical and challenging at the same time. As opposed to less-resourced languages like Persian, there are several text corpora in dominant languages like English which can be used for NLP applications.

In this paper, MirasText which is an automatically generated text corpus for Persian language is presented. In this study, over 250 Persian websites were crawled and several fields like content, description, keywords, title, etc have been extracted to generate MirasText. Topic modeling and language modeling are used to validate the generated corpus. MirasText has over 2.8 million documents and over 1.4 billion tokens, which to our knowledge is the largest Persian corpus currently available.

Keywords: Natural Language Processing, Computational Linguistics, Corpus

1 Introduction

Natural language processing (NLP) is a field of artificial intelligence and computational linguistics. It is mainly concerned with the interactions between computers and human beings. Many machine learning approaches have been proposed to solve NLP problems. In recent years because of the availability of computational resources and the introduction of new promising deep learning methods, deep learning approaches have received a lot of attention to solve various NLP problems. Today there are countless online web pages containing textual resources, which can be used to train deep learning models.

There are several text corpora available for dominant languages like English. For instance, Google published Google books n-gram corpus (Lin et al., 2012) which contains more than 800 billion tokens. Another example is Amazon review dataset (He and McAuley, 2016) which contains more than 142 million reviews crawled from Amazon website. Persian language, on the other hand, is one of the less-resourced languages. Although there are some textual datasets for this language but they do not contain huge amount of documents and consequently deep leaning methods that usually require a large corpus of articles to produce valid and reliable results may not be applicable to these corpora. For instance, Hamshahri is a corpus for Persian language (AleAhmad et al., 2009) that contains about 300 thousand news articles with semantic tags. Peykare is another Persian corpus (Bijankhan et al., 2011) with more than 100 million words. Although there is enough Persian content on the internet, but the current datasets need to be enriched.

In this paper we present MirasText which is an automatically generated text corpus for Persian. We crawled more than 250 Persian websites and processed the content to generate this new dataset. MirasText contains contents from web pages with their associated metadata like keywords, description, title, etc. MirasText contains more than 1.4 billion words and can be used for a variety of NLP applications like language modeling, summarization, title generation, keyword extraction, etc.

In the rest of this paper, Section 2 provides some basic information about MirasText. The generation process of this new corpus is discussed in Section 3. In section 4 we provide some statistics about different fields of the corpus and its semantics. Corpus validation and experimental results are presented in section 5. The conclusion is presented in Section 6, at the end of this paper.

2 Corpus Description

MirasText is a text corpus which is about **15.3** gigabytes. It is formatted as a standard CSV file, in which each line represents a document. Each document has 6 fields which are described in table 2

Field Name	**Description**
Content	web-page main content
Summary	content summary
Keywords	content keywords
Title	content title
Website	base website
URL	exact URL of the web-page

Table 1: Corpus description

Each line in MirasText contains a document with 6 fields. We had to separate each field with a delimiter that is unique enough to ensure soundness of the CSV file. The delimiter used here is three stars (i.e. ***).

MirasText is free to use for both research and business purposes and is available for download at Miras-Tech website[1].

3 Corpus Generation

The generation process of the corpus is discussed in this section. The overall architecture is illustrated in Figure 1. Crawler uses a seed of website links to explore the web, from which the results are then passed to duplicate extractor (remover) which removes the duplicate contents and writes

[1] www.miras-tech.com

the unique contents in the database. A preprocessing phase is then applied to the crawled contents to generate the final corpus. We used state of the art technologies in each part of the system. In the reminder of this section, we will explain each module in detail.

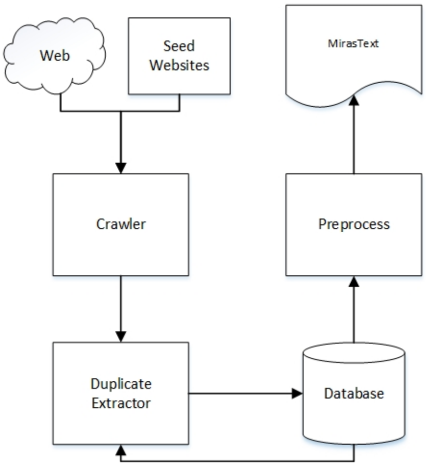

Figure 1: The system architecture used for generation of MirasText.

3.1 Seed Websites

We used a set of 250 websites to generate MirasText. These websites are selected from a wide range of scopes to ensure the diversity of corpus. Seeded websites are summarized based on their scopes in Figure 2. As shown in Figure 2, a large fraction of websites are news agencies which contain any kind of content and cannot be categorized as one of the main classes. We further analyzed corpus content to explore corpus distribution on classes in section 4.1

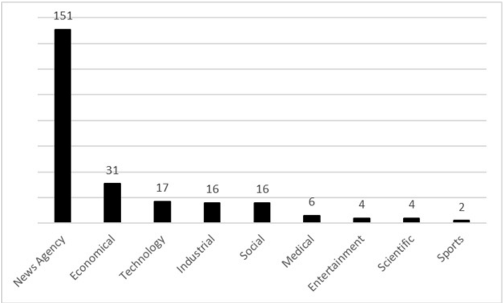

Figure 2: The scopes of initial website seeds used for crawling.

3.2 Crawler

As mentioned earlier, dataset used in this research is comprised of millions of pages crawled from hundreds of Iranian websites (mainly news agencies). As for crawler, a well-known stream processing and crawling technology is used i.e. Storm Crawler (based on Apache Storm stream processing platform (Evans, 2015)). Given a set of valid URLs, Storm Crawler fetches contents of these pages and processes the contents using a set of actions such as simple text parsing, URL extraction, keyword extraction, etc. The main steps of the crawling process are as follows

1. HTML pages are fetched as raw data then the links inside these pages are extracted for further crawling.

2. The contents of retrieved pages are parsed in order to extract useful information such as texts and keywords associated with them.

3. The last step is to index (store) the extracted fields associated with every page into a database and select a new set of URLs to continue crawling.

A high level overview of the crawling process is shown in Figure 3.

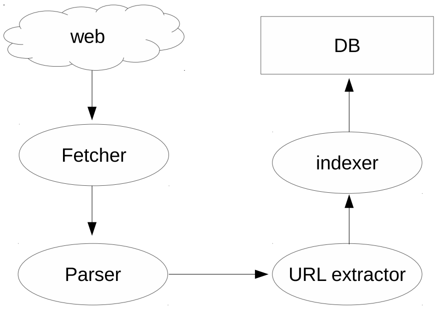

Figure 3: The overview of crawling process.

3.3 Duplicate Removal

Through the crawling process there is a possibility that certain pages with the same texts are encountered on a website. The reason for this is the ambiguity of categories in which a text could relate to, for instance a text could be categorized as *political* and *economical* simultaneously and so there would be 2 copies of it with different URLs on a website (in political and economical subsections of a web site). To remove duplicate pages from the corpus, a filtering process is used based on a bloom filter. A bloom filter (Almeida et al., 2007) is a probabilistic data structure for checking if an element is the member of a set efficiently in terms of memory used. It can also be used for removing duplicate entities from a data set by filtering out the unique ones.

4 Corpus Statistics

In this section some statistics about corpus documents and semantics is illustrated. MirasText has more than 2.8 million documents where each document has several fields. Content is the main field in each document that contains the article in the corresponding URL. Total words in the content field is more than 1.4 billion. Not only MirasText is the largest text based dataset in Persian language but also it is competitive to similar corpora in other languages like English in the sense of volume and variety. Table 2 summarizes some statistics about MirasText.

MirasText has been crawled from a wide variety of websites each containing articles with different sizes. This makes the contents length varying in a wide range of sizes from 10

Total Documents	2,835,414
Total Content Words	**1,429,878,960**
Average Content Length	504.3
Average Keywords	8.4
Average Description Length	19.8
Average Title Length	9.5

Table 2: Corpus statistics

words up to 15000 words long. Figure 4 is a histogram of content length which describes the distribution of contents on different lengths. It can be seen that most of the contents have up to 1000 words.

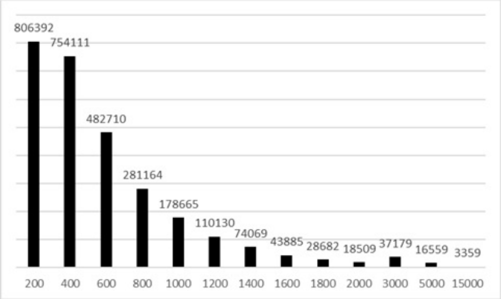

Figure 4: The histogram of content length in terms of words.

4.1 Topic Modeling

In order to explore corpus content, topic modeling is employed. It is practically impossible to manually check every document due to the size of the corpus, so we decided to use Latent Dirichlet Allocation (LDA) (Blei et al., 2003). LDA is a generative model which describes each document as a distribution on topics and describes each topic as a distribution on words. MirasText is modeled using Mallet[2] implementation of LDA which results in ***T*** and ***D*** matrices. ***T*** is a L*W matrix where L is the number of topics and W is the number of unique words. The distribution of the i'th topic on unique words is represented in the i'th row of ***T***. ***D*** is a N*L matrix where N is the number of documents. Each row in ***D*** represents the distribution of each document on all topics. Each topic is manually categorized into a set of predefined classes using its distribution on words. Knowing the distribution of documents on topics and assigning each topic to a class, we can classify each document as well. This method will result in each document being distributed on the set of predefined classes. We then averaged all documents' distributions to generate Figure 5 which gives us some insight about the contents of the corpus . It is worth mentioning that Figure 5 is just an estimation because it is generated in a semi-supervised manner.

5 Corpus Validation

MirasText is crawled automatically and needs to be validated which is done using word representation learning.

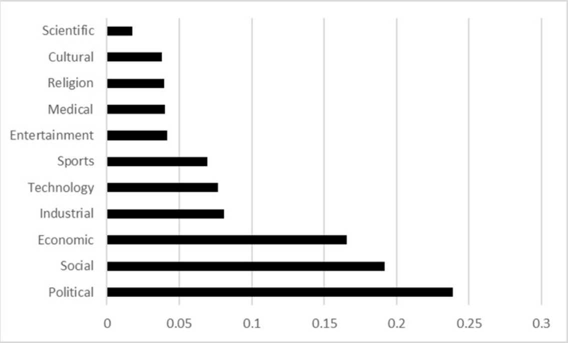

Figure 5: The distribution of contents.

word2vec is a word representation learning model that tries to code each word in a fixed length vector using the context of that word (Mikolov et al., 2013). More concretely, given a set of sentences, *word2vec* converts each unique word to a vector that represents the semantics of that word. This means that after training *word2vec* on a large and valid dataset, words with similar meanings will end up having similar vector representations. In order to validate this new corpus, *word2vec* is trained on MirasText, if the generated word representations are in fact close in the case of semantically similar words, we can conclude that MirasText data is in fact coherent and valid.

Google has published ***Google N-gram*** which contains more than 800 billion tokens from Google books (Lin et al., 2012). We trained *word2vec* model on Google N-gram corpus and used the results as a base line to validate our corpus. The overall steps of our experiment are as follows:

1. *word2vec* model is trained on MirasText and Google N-gram.

2. 1000 Persian words from a wide variety of fields are selected to form a seed list. This list is then converted to English to give us the equivalent seed list in English.

3. Word2vec model trained on MirasText is used to find 20 similar words for each word in Persian seeds where each word cluster is denoted by
$$M_i, i = 1 : 1000.$$

4. Similarly, word2vec model trained on Google N-gram is used to find 20 similar words for each word in English seed. Each word cluster is denoted by
$$G_i, i = 1 : 1000.$$

5. Finally, generated word clusters are compared to see how much they overlap. We consider word2vec model trained on Google N-gram being the best word representation model possible as it is trained on more than 800 billion tokens. If we get similar word clusters from our 1.4 billion tokens, then we can claim MirasText being as valid and coherent as Google N-gram.

[2] http://mallet.cs.umass.edu/

English word	Google N-gram cluster	Persian word	MirasText cluster
Doctor	Doctors, surgeon, dentist, pharmacist, nurse, psychologist, oncologist, gynecologist, physician, cardiologist	پزشک	دندانپزشک، پزشکان، روانپزشک، بیمار، جراح داروساز، ماما، ویزیت، پرستار، ارتوپد
Brother	Brothers, cousin, uncle, father, nephew, son, younger, sibling, dad, twin	برادر	خواهر، برادرها، عمو، کوچک، خواهرزاده پسرعمو، برادرزاده، پدر، مادر، باجناق
Football	Sports, baseball, athletics, baseball, soccer, league, coach, athletic, rugby, team	فوتبال	هندبال، والیبال، فوتسال، بسکتبال، لیگ، مربی بازیکن، باشگاه، جودو، فوتبالی

Table 3: The sample equivalent word clusters extracted from MirasText and Google N-gram

5.1 Experimental Results

We conducted evaluations according to the general measuring method used in the Information retrieval evaluation, i.e. precision (P), recall (R) and F1-Measure. The evaluation measures are defined as follows:

$$P = \frac{\sum_{i=1}^{1000} |G_i \cap M_i|}{\sum_{i=1}^{1000} |M_i|} \tag{1}$$

$$R = \frac{\sum_{i=1}^{1000} |G_i \cap M_i|}{\sum_{i=1}^{1000} |G_i|} \tag{2}$$

$$F1 = \frac{2PR}{P+R} \tag{3}$$

We used the Gensim implementation [3] of *word2vec* for training. Each generated model is fed with the corresponding Persian or English seed words to generate word clusters G_i and M_i (for i=1 to 1000). It is worth mentioning that computing precision, recall and F1-Measure needs to be done manually because each G_i and M_i are word clusters in English and Persian respectively, and computing their overlap needs translation. Although this translation could be done automatically, we preferred to do this manually in order to get more reliable results. Table 3. illustrates three cherry-picked word clusters generated by *word2vec* on Google N-gram and MirasText.

After manually computing the evaluation measures, we get a F1-Measure of ***0.75*** which indicates high correlation between extracted clusters from Google N-gram and MirasText. This experiment represents the validity and coherence of MirasText as *word2vec* could learn a reliable word representation from it.

6 Conclusion

In this paper, Miras-Text which is an automatically generated text corpus for Persian language, is presented. The system developed in this study uses a list of websites to generate a text corpus. This system crawls the specified websites and extracts useful information like content, description, keywords, title, etc. The generated corpus contains more than 2.8 million documents and more than 1.4 billion content words. MirasText is the largest Persian text corpus available which can be used for a variety of NLP applications like language modeling, automatic summarization, keyword extraction, title generation, etc. In order to validate the coherence of Miras-Text, a word2vec model is trained both on MirasText and Google N-gram corpus. The trained models are then used to generate some word clusters. The comparison of these word clusters shows high correlation between the two models which indicates the validity and coherence of MirasText.

7 Acknowledgements

The authors would like to thank everyone at Miras-Tech especially the members of team Pegasus who put a lot of effort into building the presented corpus.

8 References

AleAhmad, A., Amiri, H., Darrudi, E., Rahgozar, M., and Oroumchian, F. (2009). Hamshahri: A standard persian text collection. *Knowledge-Based Systems*, 22(5):382–387.

Almeida, P. S., Baquero, C., Preguiça, N., and Hutchison, D. (2007). Scalable bloom filters. *Information Processing Letters*, 101(6):255–261.

Bijankhan, M., Sheykhzadegan, J., Bahrani, M., and Ghayoomi, M. (2011). Lessons from building a persian written corpus: Peykare. *Language resources and evaluation*, 45(2):143–164.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Evans, R. (2015). Apache storm, a hands on tutorial. In *Cloud Engineering (IC2E), 2015 IEEE International Conference on*, pages 2–2. IEEE.

He, R. and McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 507–517. International World Wide Web Conferences Steering Committee.

Lin, Y., Michel, J.-B., Aiden Lieberman, E., Orwant, J., Brockman, W., and Petrov, S. (2012). Syntactic annotations for the google books ngram corpus. In *Proceedings of the ACL 2012 System Demonstrations*, pages 169–174. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[3] radimrehurek.com/gensim/models/word2vec.html