Word Embeddings Go to Italy: a Comparison of Models and Training Datasets

Giacomo Berardi, Andrea Esuli, Diego Marcheggiani

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"
Consiglio Nazionale delle Ricerche
via Giuseppe Moruzzi, 1, 56124 Pisa, Italy
firstname.lastname@isti.cnr.it

Abstract. In this paper we present some preliminary results on the generation of word embeddings for the Italian language. We compare two popular word representation models, `word2vec` and GloVe, and train them on two datasets with different stylistic properties. We test the generated word embeddings on a word analogy test derived from the one originally proposed for `word2vec`, adapted to capture some of the linguistic aspects that are specific of Italian.
Results show that the tested models are able to create syntactically and semantically meaningful word embeddings despite the higher morphological complexity of Italian with respect to English. Moreover, we have found that the stylistic properties of the training dataset plays a relevant role in the type of information captured by the produced vectors.

Keywords: word2vec, glove, word embeddings

1 Introduction

Research on word representation models [19,20], *word embeddings*, has gained a lot of attention in the recent years [9]. This happened also thanks to a renewed boost in neural network technologies, *deep learning*, which have been successfully applied to this task. The most popular of this series of work is without doubt Mikolow's `word2vec` [15,16], which can be considered as a turning point in the field of word representation. `Word2vec` uses a "shallow" neural network that can easily process billions of word occurrences, and create semantically and syntactically meaningful word representations in few hours, in contrast to previous models [9] that may need weeks to train.

Distributed word representations, also known as word embeddings, represent a word as a vector in $\mathbb{R}^n$. The more the vectors are closer the more the two corresponding words are deemed to share some degree of syntactical or semantical similarity. Word embeddings are typically obtained as the by-product of the training of a neural language model [6,10] that learns to predict the most probable word given a set of context words. Word embeddings have empirically proved to be helpful in the solution on several tasks that span from NLP [17,21] to information retrieval [3,8].

In this work we compare the performance on the Italian language of two widely adopted "shallow" word representation models, `word2vec` and GloVe [18]. We train them on two datasets that have different stylistic properties, with the aim of checking if such differences have an impact on the produced embeddings. We have produced an Italian version of the Google word analogy test in order qualitatively compare the performance of the different models, we make it publicly available. We also make publicly available the word vectors obtained by the above-mentioned models on the two datasets[1].

2 Related Work

Despite the flurry of work in recent years on word representation models, few works have studied the adaptation of these models in languages different from English. The Polyglot project [1] produced word embeddings for about 100 languages by following the model proposed by [9] and Wikipedia as the source of text. Attardi has been among the firsts to produce word representations in Italian[2] by using the Italian Wikipedia, adopting the deep learning model proposed in [9], and the code[3] used in [1]. Attardi and Simi [2] used such word embeddings as additional input features for a dependency parser, but did not observed improvements over a baseline system not using such features. Basile and Novielli [4] trained `word2vec` on a set of 10 million tweets, and used the resulting vectors to measure the similarity of tweets with respect to prototype vectors of sentiment labels (positive, negative, mixed, neutral). The use of such similarity features produced an improvement of 6.71% and 2.61% respectively on tweet subjectivity and polarity classification. Cimino et al. [7] trained `word2vec` on the PAISÁ corpus [14] (388,000 documents, 250 million tokens) and on a set of 1.2 million tweets. They built two distinct word embeddings models on which they computed word similarity as the cosine similarity between word vectors. Lomonaco [13] presented online results of a demo test of `word2vec`, using a small Italian training set (33 million tokens) and a direct translation of the Google analogy test, obtaining low accuracy values, less than 8%.

3 Algorithms

The generation of word embeddings is typically obtained as the by-product of the training of (neural) language models. In these models a word embedding is a vector in $\mathbb{R}^n$, with the value of each dimension being a feature that weights the relation of the word with a "latent" aspect of the language. These features are jointly learned with other weights/parameters of the model from plain, unannotated text, according to an objective function that typically recalls the distributional hypothesis, i.e., that words frequently used in similar contexts have similar meaning. See [5] for an overview on representation learning.

[1] `http://hlt.isti.cnr.it/wordembeddings`
[2] `http://tanl.di.unipi.it/embeddings/`
[3] `https://bitbucket.org/aboSamoor/word2embeddings`

The two methods for word representation we compare in this paper are the Skip-gram model of `word2vec`, and GloVe.

3.1 `word2vec`

Mikolov et al. [15] investigated on two different models that seek to optimize two objective functions that aim at maximizing respectively the probability of a word given its context (*Continuous bag-of-word* model) and the probability of the surrounding words (before and after the current word) given the current word (*Skip-gram* model). In this paper we use the Skip-gram model, which obtained better results than the continuous bag-of-word model [16].

The Skip-gram algorithm models the probability distribution of a context word w_c, given a word w_i. The context-wise probability to maximize is:

$$p(w_c|w_i) = \frac{\exp(\mathbf{u}_{w_c}^T \mathbf{v}_{w_i})}{\sum_{c'=1}^{C} \exp(\mathbf{u}_{w_{c'}}^T \mathbf{v}_{w_i})} \tag{1}$$

where $\mathbf{u}_{w_c}$ is the output context vector associated to the context w_c, and $\mathbf{v}_{w_i}$ is the input word vector (i.e., the word embedding) associated to the word w_i, C is the set of all available contexts.

The computation of Equation (1) is impractical for big datasets. For this reason Mikolov et al. introduced in [16] a negative sampling approximation where the denominator is canceled out from the training computation. The size of the context set and the size of the parameter vectors $\mathbf{v}_{w_i}$ and $\mathbf{u}_{w_c}$ (i.e., the length of the word embeddings) are set by the user.

3.2 GloVe

GloVe [18] (*Glo*bal *Ve*ctors) sequentially analyzes word contexts iterating on word windows across the corpus. The authors define $P_{ij} = p(j|i)$ as the probability that the word w_j appears in the context of word w_i. The authors then define the ratio P_{ic}/P_{jc} as a measure to study the relationship between two words w_i and w_j, given a set of context words w_c. The ratio is large for context words related to w_i and small for context words related to w_j, while it is close to one when the context word is related to both words.

This ratio is used in the objective function:

$$F(\mathbf{v}_{w_i}, \mathbf{v}_{w_j}, \mathbf{u}_{w_c}) = P_{ic}/P_{jc} \tag{2}$$

The vector representation of words $\mathbf{v}_{w_i}$, $\mathbf{v}_{w_j}$ (i.e., the word embeddings), and contexts $\mathbf{u}_{w_c}$ are the parameters to learn in the function F, which returns a scalar (the above defined ratio) for each combination of i, j and c. The authors derive a function F which encodes the relationship between the three words, and they cast the equation as a least squares problem, so to train the model using a gradient descent algorithm. The final cost function to minimize is:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(\mathbf{v}_{w_i} \cdot \mathbf{u}_{w_j} + b_i + b_j - logX_{ij})^2 \tag{3}$$

Table 1. Datasets statistics.

	Documents	Sentences	Vocabulary	Vocabulary $tf \geq 5$	Tokens
WIKI	3,521,118	9,483,245	6,246,253	733,392	311,874,402
BOOKS	31,432	232,777,927	9,443,100	1,910,809	2,222,726,367

which is a drastic simplification over Equation (2). Here V is the word vocabulary, X_{ij} is the number of times the word w_j appears in the context of word w_i, b_i and b_j are bias terms, f is a weighting function that cuts off low co-occurrences, which are usually noisy, and also avoiding to overweight high co-occurrences.

4 Experiments

4.1 Dataset

We trained our models on two datasets in the Italian language: the entire dump of the Italian Wikipedia (WIKI, dated 2015/02/24) and a collection of 31,432 books[4] (mostly novels) written in Italian (BOOKS). The purpose and style of the two datasets is very different. WIKI's content aims at convey knowledge with an encyclopedic style, in which a simple but verbose language is preferred. BOOKS's content mostly aims at entertaining the reader, using a rich and complex language, including dialogues and first-person speech – which are almost missing in WIKI – but usually is not interested in providing the user with detailed and rigorous knowledge about the world. WIKI's sentences are on average composed of 32 words, while BOOK's sentences are on average composed of just nine words. This difference will help to investigate the impact of the training data domain for the creation of semantically significant word embeddings.

In order to evaluate the created models we have manually translated the Google word analogy test for English. The original test[5] is composed by 19,558 questions divided in semantic questions e.g., *father : mother = grandpa : grandma*, and syntactic questions e.g., *going : went = predicting : predicted*. We have started by translating the English test to Italian. In this work we explore only single-word embeddings, and this led us to make a few changes for some syntactic categories of the test. For example, the Italian version the `gram3-comparative` section is limited to a few adjectives that have a single-word comparative version, given that in Italian comparatives are usually built as multi-word expressions (*smart : smarter = intelligente : piú intelligente di*). Similarly, the `gram4-superlative` section has been translated to Italian absolute superlatives, which are usually expressed by a single word, rather than relative superlatives,

[4] This collection is the result of a semi-automated crawling of various websites that publish free e-books. Copyright policies do not allow us to redistribute the books. We are working on alternative ways to make this collection replicable.

[5] `http://word2vec.googlecode.com/svn/trunk/questions-words.txt`

which are usually multi-word expressions. The section `gram5-present-participle` of English has been mapped to the Italian gerund, as it is of more common use in Italian. For the section `gram9-plural-verbs` we split it in two sections, one using the third person, and one using the first person. Given the higher complexity of Italian with respect to verb tenses and the different suffixes use to determine number and gender, we have added three more sections dedicated to these aspects: `present-remote-past-verbs (1st person)`, `masculine-feminine-singular` and `masculine-feminine-plural`.
We also added a new category of questions among the semantic questions called `regione-capoluogo`, which focuses on Italian geographic information. The Italian test consists of a total of 19,791 questions.

4.2 Parameters Setting

For our experiments we used the Python implementation of `word2vec` provided by Gensim[6] and the C implementation shared by the creators of GloVe[7]. We set `word2vec` parameters following the best performing values reported on [16]: we used negative sampling with 10 negative samples and a context window size of 10 words. We set at 300 the size of the word embeddings vector. For GloVe we choose the default values as reported in [18], apart from the context window size and vector size, which we set respectively to 10 and 300, to match `word2vec`. We also tested the pre-trained vectors released by the Polyglot project[8], which are trained on a dump of the Italian Wikipedia dated 2013/07/21.

4.3 Results

We determined the answers for the word analogy test as in [16], selecting the word whose vector maximizes the formula (dubbed Cos3Add in [11]):

$$\arg\max_{b^* \in V}(cos(b^*, b - a + a^*)) \tag{4}$$

where b^* is the word to be guessed (e.g., "king"), b is its know paired word ("man"), and a, a^* are the known pair ("woman","queen").

We have measured the accuracy in solving the test as the ratio between the number correct answers and the total number of questions in the test, considering as wrong answers also the cases in which the words were not in the model, usually because it did not appear with enough frequency in the training dataset. This kind of evaluation allows to compare results over different datasets, penalizing those datasets in which many words are missing.

A first observation from the results of Table 2 is that overall the performance of the Skip-gram model on Italian is quite good, but not as good as the performance it obtained on English in [16], $\sim$ 47% accuracy on the Italian versus

[6] `http://radimrehurek.com/gensim/`
[7] `http://nlp.stanford.edu/projects/glove/`
[8] `https://sites.google.com/site/rmyeid/projects/polyglot`

$\sim 60\%$ accuracy on the English. The results, even though not directly comparable, may be a sign of a higher complexity of Italian with respect to English.

A second interesting observation is the different performance of the models when trained on WIKI or on BOOKS. For the Skip-gram model, WIKI is about 10% more accurate on semantic questions, BOOKS is instead 20% more accurate on syntactic questions. GloVe has this difference only on the syntactic questions, but is still a relevant 14%. We attribute this behavior to the different aims and styles of the two datasets: the first one more encyclopedic and notion-centric, the second one more narrative and rich of syntactic variations i.e., more verb tenses, more declinations. This result is important if we think about how the word embeddings are used in practical applications. For example one could benefit from semantic-rich vectors for query expansion, while syntax-rich vectors can be beneficial in parsing tasks.

The overall performance of GloVe is significantly worse than the Skip-gram model. This is in contrast with the results on English reported in [18]. Further studies are necessary to understand if the obtained results are due to the intrinsic complexity of Italian or to a poor choice of GloVe parameters.

Polyglot results are much worse than the other tested methods. This is probably due to the small vocabulary size of Polyglot and the small embedding size (respectively 100,000 and 64 as reported in [1]). Polyglot skipped about 33% of the questions due to missing words in the models, while the other models skipped less than 10% of them. In order to have a fair comparison, for the future we plan to produce Polyglot-like vectors using our datasets and larger parameters values, similar to those used for the other methods.

5 Conclusion

We have tested two popular word reprentation methods, `word2vec`'s Skip-gram and GloVe, training them on two datasets. Skip-gram obtained the best results over the other methods. The literature is rich of many other proposals of word representation models, we plan to expand our comparison in the future, e.g., to `word2vec`'s Continuous bag-of-word algorithm, the SPPMI and SPPMI-SVD models introduced in [12].

The different stylistic properties of the two datasets had an impact on the semantic/syntactic properties of the generated vectors. Future work will investigate on which kind of dataset is the most appropriate for the solution of a given task. For example, a query expansion task may benefit from semantic-rich embeddings, while a parsing task may benefit from syntax-rich embedding.

We have adopted a simple word analogy test to evaluate the generated word embeddings. This test has pros and cons with respect to performing an extrinsic evaluation by putting the word embeddings at work in some practical application, e.g., text classification, named entity recognition. The positive aspects of this testing methodology are the speed of execution, the easiness of inspection, the possibility of crafting tests to focus on a specific linguistic aspect (singular vs plural, male vs female...). The main negative aspect is that the observed

Table 2. Accuracy results on the analogy test. *Polyglot vectors are those computed in [1] on a different Wikipedia dump.

	w2v-Skip-gram		GloVe		Polyglot*
	WIKI	BOOKS	WIKI	BOOKS	WIKI*
capital-common-countries	87.55%	84.78%	66.40%	61.07%	5.93%
capital-world	63.86%	47.35%	22.74%	21.89%	0.90%
currency	5.31%	3.58%	1.27%	0.58%	0.12%
city-in-state	29.19%	23.47%	11.76%	15.04%	1.34%
regione-capoluogo	41.23%	23.10%	23.10%	16.08%	1.75%
family	58.01%	67.98%	51.44%	59.58%	29.92%
accuracy on semantic	48.81%	38.56%	21.33%	21.54%	2.50%
adjective-to-adverb	12.58%	17.74%	11.51%	14.52%	1.40%
opposite	7.43%	27.54%	8.15%	25.91%	2.17%
comparative	0.00%	8.33%	8.33%	8.33%	8.33%
superlative (absolute)	8.72%	48.03%	16.83%	34.03%	1.47%
present-participle (gerund)	55.02%	77.84%	51.89%	78.50%	9.19%
nationality-adjective	77.36%	77.30%	68.17%	52.47%	2.13%
past-tense	19.60%	61.36%	32.39%	63.64%	1.89%
plural	40.79%	54.21%	31.67%	50.40%	4.60%
plural-verbs (3rd person)	72.20%	85.73%	54.98%	74.29%	33.70%
plural-verbs (1st person)	0.54%	45.05%	0.11%	30.32%	0.22%
present-remote-past-verbs (1st person)	0.43%	34.41%	0.11%	18.60%	0.00%
masculine-feminine-singular	35.71%	57.58%	33.12%	41.13%	9.52%
masculine-feminine-plural	4.11%	29.00%	3.03%	15.37%	0.43%
accuracy on syntactic	32.62%	54.56%	30.20%	44.60%	5.23%
overall accuracy	39.91%	47.35%	26.21%	34.21%	4.00%

differences are not guaranteed to generate a similar difference in the final result produced by the applications in which such word embeddings will be used. Future work will include testing the word embeddings in practical applications.

References

1. Al-Rfou, R., Perozzi, B., Skiena, S.: Polyglot: Distributed word representations for multilingual NLP. In: Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL 2013). pp. 183–192. Sofia, BG (2013)
2. Attardi, G., Simi, M.: Dependency parsing techniques for information extraction. In: 4th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2014). pp. 9–14. Pisa, IT (2014)
3. Baeza-Yates, R.A., Jiang, D., Silvestri, F., Harrison, B.: Predicting the next app that you are going to use. In: Proceedings of the 8th ACM International Conference on Web Search and Data Mining (WSDM 2015). pp. 285–294. Shanghai, CH (2015)
4. Basile, P., Novielli, N.: Uniba at evalita 2014-sentipolc task: Predicting tweet sentiment polarity combining micro-blogging, lexicon and semantic features. In: 4th

evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2014). pp. 58–63. Pisa, IT (2014)
5. Bengio, Y., Courville, A.C., Vincent, P.: Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35(8), 1798–1828 (2013)
6. Bengio, Y., Schwenk, H., Senécal, J.S., Morin, F., Gauvain, J.L.: Neural probabilistic language models. In: Innovations in Machine Learning, pp. 137–186 (2006)
7. Cimino, A., Cresci, S., DellOrletta, F., Tesconi, M.: Linguistically–motivated and lexicon features for sentiment analysis of italian tweets. In: 4th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2014). pp. 81–86. Pisa, IT (2014)
8. Clinchant, S., Perronnin, F.: Aggregating continuous word embeddings for information retrieval. In: Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality. pp. 100–109. Sofia, BG (2013)
9. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research 12, 2493–2537 (2011)
10. Goldberg, Y., Levy, O.: word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722 (2014)
11. Levy, O., Goldberg, Y.: Linguistic regularities in sparse and explicit word representations. In: Proceedings of the 18th Conference on Computational Natural Language Learning, (CoNLL 2014). pp. 171–180. Baltimore, US (2014)
12. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: 27th Annual Conference on Neural Information Processing Systems (NIPS 2014). pp. 2177–2185. Montreal, CA (2014)
13. Lomonaco, V.: Word2vec on the italian language: first experiments. `http://goo.gl/x57luD`, accessed: 2015-03-12
14. Lyding, V., Stemle, E., Borghetti, C., Brunello, M., Castagnoli, S., DellOrletta, F., Dittmann, H., Lenci, A., Pirrelli, V.: The paisa corpus of italian web texts. In: Proceedings of the 9th Web as Corpus Workshop (WaC-9). pp. 36–43 (2014)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at International Conference on Learning Representations (ICLR 2013). Scottsdale, US (2013)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of Advances in Neural Information Processing Systems (NIPS 2013). pp. 3111–3119. Lake Tahoe, US (2013)
17. Passos, A., Kumar, V., McCallum, A.: Lexicon infused phrase embeddings for named entity resolution. In: Proceedings of the 18th Conference on Computational Natural Language Learning (CoNLL 2014). pp. 78–86. Baltimore, US (2014)
18. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014). pp. 1532–1543. Doha, QA (2014)
19. Sahlgren, M.: The Word-Space Model. Ph.D. thesis, Department of Linguistics, Stockholm University (2006)
20. Schütze, H.: Word space. In: Advances in Neural Information Processing Systems (NIPS 1992). pp. 895–902. Denver US (1992)
21. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th annual meeting of the association for computational linguistics (ACL 2010). pp. 384–394. Uppsala, SE (2010)