# Local-Global Vectors to Improve Unigram Terminology Extraction

**Ehsan Amjadian[1,2], Diana Inkpen[2], T. Sima Paribakht[3], and Farahnaz Faez[4]**
[1]Institute of Cognitive Science, Carleton University, Canada
[2]School of Electrical Engineering and Computer Science, University of Ottawa, Canada
[3]Official Languages and Bilingualism Institute, University of Ottawa, Canada
[4]Faculty of Education, Western University, Canada.
ehsan.amjadian@carleotn.ca, diana@site.uottawa.ca
sima.paribakht@uottawa.ca, ffaez@uwo.ca

## Abstract

The present paper explores a novel method that integrates efficient distributed representations with terminology extraction. We show that the information from a small number of observed instances can be combined with local and global word embeddings to remarkably improve the term extraction results on unigram terms. To do so, we pass the terms extracted by other tools to a filter made of the local-global embeddings and a classifier which in turn decides whether or not a term candidate is a term. The filter can also be used as a hub to merge different term extraction tools into a single higher-performing system. We compare filters that use the skip-gram architecture and filters that employ the CBOW architecture for the task at hand.

## 1 Introduction

The terminology of a domain encodes the existing knowledge in that domain. Hence understanding and interpreting a message belonging to a domain cannot be fully achieved without knowing its terminology. This makes Automatic Terminology Extraction (ATE) an important task in Natural Language Processing (NLP). ATE methods have been conventionally classified as linguistic, statistical, and hybrid (Cabre-Castellvi et al., 2001; Chung, 2003). Linguistic methods implement formal rules to detect terms; statistical methods exploit some measures based on relative frequency of terms in general and target corpora by means of which they can tell apart a term from a word in its generic sense; and, the hybrid methods combine the advantages of both of these techniques (Frantzi et al., 1998; Park et al., 2002; Drouin, 2003; Chung and Nation, 2004; Yoshida and Nakagawa, 2005; Vu et al., 2008; VRL, 2009; Yang et al., 2010; Zervanou, 2010; Broß and Ehrig, 2013; Conrado et al., 2013). These methods often regard words in a document as atomic elements; that is, they are manifested as their symbolic alphabetical form in the algorithm (such as in 'a' below) and/or as some measure of their relative frequency (as in 'b' below). But, in a distributed approach (as in 'c' below) each word has tens or hundreds of real-valued components, as opposed to a single linguistic form or a termhood score[1]. The idea is that such finer-granularity may grant more access to the information that a word contains, potentially resulting in a better detection of terms in a document.

(a) apple

(b) 0.0003654

(c) [0.54407, 0.9233, 0.50644, 0.46454, -0.62015, -0.35166, ... -0.93253]$_n$

where $n$ is often between 50 and 1000 for different types of word embeddings, almost similar to other vector space models such as Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). In addition to the richer representation, the rise of distributed methods in NLP, especially the recent word embeddings surge, makes it relevant to explore the ways current model architectures may fit into the

[1] Termhood is the degree of a linguistic unit being related to a domain-specific concept (Kageura and Umino, 1996)

*Proceedings of the 5th International Workshop on Computational Terminology,*
pages 2–11, Osaka, Japan, December 12 2016.

automatic terminology extraction picture. However, the fact that makes them particularly appealing is their computational efficiency and scalability as compared to the available alternatives, including LSA and LDA (Mikolov et al., 2013).

We present a simple method that harnesses the rich distributed representation acquired by a log-bilinear regression model called GloVe (Pennington et al., 2014), as well as the efficiency of log-linear models with CBOW[2] and skip-gram architectures (Mikolov et al., 2013), which is a step forward towards language-independent ATE. In our method, the GloVe model is used to preserve the global[3] scope of a word in a general corpus (i.e., its general sense(s)) and the CBOW or the skip-gram model is used to capture the local scope for a word in a technical corpus (i.e., its technical usage).

Currently, we use our method as a filter on top of a previously-developed hybrid term extraction algorithm, namely, TermoStat (Drouin, 2003; Serrec et al., 2010) along with two simpler methods (refer to section 4 for further details) and focus on the unigram[4] term extraction. TermoStat has been previously tested on mathematics domain where it performed well on the extraction of multi-word expressions, but lower on unigram terms (Inkpen et al., 2016), hence the present work is an attempt to improve unigram term extraction for the same domain.

As mentioned, the target domain of the present study is mathematics textbooks. A significant component of any academic subject is its terminology. Knowledge of the terminology of a field enables students to engage with their discipline more effectively by enhancing their ability to understand the related academic texts and lectures, and allowing them to use the subject-specific terminology in their discussions, presentations, and assignments. Therefore, generating lists of terms specific to various fields of study is a significant endeavor. However, these lists have often been generated manually or through corpus-based studies, which are time consuming, labor-intensive, and prone to human error. This can be facilitated by a great extent with high-performance automatic term extraction.

To the best of our knowledge, the present method is the first to successfully apply neural network word embeddings to the terminology extraction task. This method can be combined with any term extraction algorithm for any non-polysynthetic[5] language and any domain.

## 2 Related Work

As described in section 1, ATE approaches traditionally fall into three categories, namely, linguistic, (unsupervised) statistical[6], and hybrid methods (Cabre-Castellvi et al., 2001; Chung, 2003). These TE methods have been applied to both monolingual and multilingual corpora (Ljubešić et al., 2012). Linguistic methods apply hand-coded rules to the target corpus to extract technical terms. Statistical methods are often unsupervised and apply some measure of relative frequency to a technical target corpus, a reference (general) corpus and sometimes a (contrastive) corpus from another domain, to identify the existing terms in the technical corpus (Frantzi et al., 1998; Chung, 2003; Vu et al., 2008; Conrado et al., 2013). Hybrid methods combine statistical and linguistic methods to extract terminology from a target corpus and often perform well (Drouin, 2003; Serrec et al., 2010; Ismail and Manandhar, 2010; Vintar, 2010). The above-mentioned approaches, in contrast to the method put forth in this paper, regard words as atomic units represented by their linguistic forms and their statistical scores that indicate their likelihood to be terms. They may, however, implement rules associated with some linguistic features (e.g., their POS tags, their position in the POS sequence, their position in the parse tree, phrase, and/or in the sentence). These linguistic rules make an algorithm language-dependent and even sometimes to some degree domain-dependent. On the contrary, our method, if used independently, can be used for any non-polysynthetic language and for any domain as long as domain-specific and general corpora are available.

[2] Continuous Bag Of Words. See section 4 for further details.

[3] We use the terms "global" and "local" with a different sense from Pennington et al. (2014). They used "global" to denote a model that captures a wider set of co-occurrence statistics being computed globally (e.g., document-wide) such as in LSA, as compared to the "local" methods that use a relatively small context window for co-occurrence computation such as CBOW and skip-gram (Mikolov et al., 2013) or similarly vLBL and ivLBL (Mnih and Kavukcuoglu, 2013).

[4] Singe-words

[5] A polysynthetic language has a richer morphology than syntax, where the words are much longer and can convey full sentence-like messages

[6] These statistical methods are distinct from statistical learning approaches.

In this study, we do not use our method independent of other methods, but it can still be regarded as a step towards a language independent ATE algorithm that benefits from latent linguistic information encoded in the vectors used (see section 4 for further details on the method) in comparison to purely statistical methods that do not capture such information.

Supervised methods have been recently[7] designed for terminology extraction. Nazar and Cabré (2012) used examples of terms from the domain of interest and a reference corpus of general language, which represent positive and negative examples of terms, and a three-level (i.e., syntactic, lexical, and morphological) learning algorithm to detect the terms. They used the frequency distribution for POS tag sequences at the syntactic level. At the lexical level, they accounted for the frequency of the lexical units within the terms (word forms, as well as lemmas). Finally, at their morphological level, from each word type they extracted initial and final character $n$-grams where: $1 \leq n \leq 5$ (Nazar and Cabré, 2012). Their term extractor is an online web-based system that is constantly being updated when used by terminologists. More recently, Conrado et al. (2013) achieved state-of-the-art performance for unigram term extraction in Brazilian Portuguese using supervised learning algorithms and a rich feature set. They used eight linguistic features, seven statistical features, and four hybrid features in their method. The present work would be the next phase for these supervised methods, since we move closer to a rich, language-independent, resource-independent, and fully data-driven representation. It is worth noting that modern word embeddings have been successfully employed in many tasks, including the related areas of keyphrase extraction (Wang et al., 2015) and aspect term extraction[8] (Yin et al., 2016); nevertheless, this is the first time they are leveraged for the general terminology extraction task.

## 3 Corpus

The domain corpus that we used for the purpose of this study is comprised of 5 English high school mathematics textbooks used in Ontario, Canada (Small et al., 2005; Small et al., 2007a; Small et al., 2007b; Kirkpatrick et al., 2007; Crippin et al., 2007), which were concatenated into a corpus consisting of 1,127,987 tokens.

## 4 Methodology

In this study, we merged the results of three tools (see below) that we used for terminology extraction. We improved the performance of these tools by adding local-global distributed word representations coupled with a classifier as a filter. The basic idea is if a candidate term in a technical corpus is being used in a distinctly different manner and context than in a general corpus, then it is likely to be a term[9]. For each Candidate Term (CT) extracted by TermoStat, two separate embedding vectors are constructed and then concatenated. One is a global vector pre-trained on general corpora, and the other is a local vector trained on the target corpus from which the terms are extracted. Each of these two vectors portrays distinct regularities about the CT at hand, as discussed below.

The idea behind using a general global vector is to encapsulate the behavior of the CT in its generic sense(s), the intuition being that the generic sense(s) have a predominant presence in general corpora and will, therefore, dominate the vector. We use the pre-built GloVe vectors as our global vectors, created by Pennington et al. (2014)[10]. These global vectors are of 50 dimensions and were built on Wikipedia 2014 + the Gigaword 5 corpus; that is, approximately 6 billion tokens. GloVe is a log-bilinear regression model. More specifically:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(\mathrm{w}_i^T \tilde{\mathrm{w}}_j + b_i + \tilde{b_j} - \log X_{ij})^2 \tag{1}$$

[7] There has, however, been earlier supervised work in keyword/keyphrase extraction such as Turney (2000), as opposed to terminology extraction which is the topic of this paper. While Keyword extraction is the task of extracting only a few keywords in a text, terminology extraction needs to detect all the terms, usually from a large domain corpus.

[8] Aspect term extraction is the task to identify the aspect expressions which refer to a products or services properties or attributes, from customer reviews (Pontiki et al., 2014; Pontiki et al., 2015; Yin et al., 2016).

[9] This is similar to the premise of traditional statistical ATE methods except that those models carry less local information such as syntactic behavior.

[10] Available at: http://nlp.stanford.edu/projects/glove/

where $V$ is the size of the vocabulary, $f(X_{ij})$ is the weighting function, $\mathrm{w}_i$ and $\mathrm{w}_j$ are two separate context word vectors and their sum constructs the final GloVe vector, and finally $b_i$ and $b_j$ are biases for their corresponding word vectors. GloVe has been shown to adequately reflect both semantic and syntactic regularities in the data (Pennington et al., 2014); we require both for our global embeddings.

In contrast to global embeddings, technical local embeddings are built on the domain corpus. These vectors are valuable since they capture the behavior of the candidate terms in the technical domain. To construct the local embeddings, we use two neural network architectures introduced by Mikolov et al. (2013) on our corpus (discussed in section 3 above), namely, the CBOW and the skip-gram architectures shown in Figure 1. CBOW and skip-gram are efficient algorithms trained by stochastic gradient descent and backpropagation. Below are their complexities, respectively:

$$Q_{CBOW} = N \times D + D \times \log_2(V) \tag{2}$$

$$Q_{skipgram} = C \times (D + D \times \log_2(V)) \tag{3}$$

where $N$ is the number of context words, $D$ the vector dimensionality, $V$ the vocabulary size, and $C$ is roughly the maximum distance for the context from the target word. CBOW is trained to predict a target word based on its surrounding words, and the skip-gram model is trained to predict the surrounding words given a single word. The CBOW architecture tends to have better performance in discovering syntactic regularities as compared to semantic regularities, whereas the skip-gram architecture tends to have a higher performance in finding semantic regularities rather than syntactic ones (Mitkov et al., 2012; Pennington et al., 2014). Because we are dealing with unigram terms and not multi-word terms at this stage, we expect a skip-gram filter to outperform a CBOW filter. We used the gensim[11] implementation (Řehůřek and Sojka, 2010) of word2vec[12] to build vectors of 100 dimensions with context window of size 5 and minimum frequency of 5. The rest of the parameters were left with their default values.

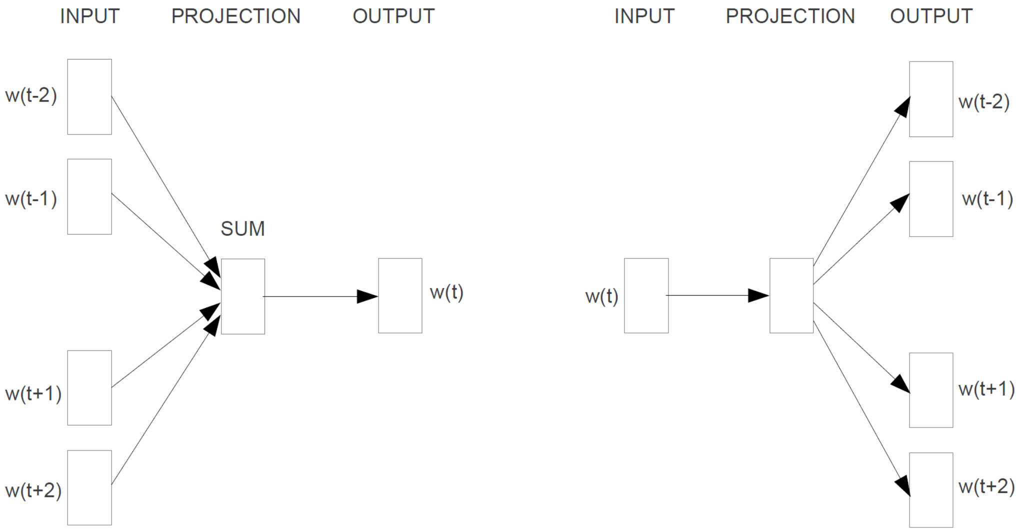

Figure 1: The CBOW architecture predicts the current word given the surrounding words, and the skip-gram predicts the surrounding words given a word (Mikolov et al., 2013).

After having the local and global vectors ready, they are concatenated and the resultant local-global vector is fed to the classifier to make the final decision. We experimented with several classification algorithms. Following Conrado et al. (2013) (see section 2 for more details), we used JRip[13], Naíve Bayes, J48[14], and SMO[15](Platt, 1998). We also tested a few other classifiers of our choice to find the

[11] Available at: https://radimrehurek.com/gensim/models/word2vec.html
[12] Available at: https://code.google.com/archive/p/word2vec/
[13] A rule induction classifier
[14] A decision tree algorithm. We used it with confidence factor of 0.25.
[15] A Support Vector Machine classifier from Weka

most suitable ones for the task, including logistic regression, multi-layer perceptron and PART[16]. Default parameters were used for these classifiers. We employed the Weka implementation of all the above-mentioned classifiers. We tested all the classifiers for both the CBOW and the skip-gram architecture.

As mentioned above, our method operates on the results of three other term extraction tools. The first is a full-fledged hybrid ATE tool called TermoStat[17] (Drouin, 2003). It statistically computes the specificity of a word in a multi-word expression with reference to a general corpus and uses POS-tag patterns to detect head nouns and term phrases. The second term extraction tool is called Topia[18]. We augmented it by a filter that removed all the candidate terms that had less than 3 letters and took out numbers or special characters from candidate terms. Topia uses the majority POS tag for each word, and applies only a frequency threshold to extract terms. Third, we extracted most frequent unigrams using AntConc[19][20] (Anthony, 2012), and filtered out all the stop-words.

Figure 2 illustrates our overall system. First, the term extraction tools operate on the target corpus. Then, the resultant TC's from all of them are pooled together (with no repetition) and fed to the filter. The filter uses the local vectors trained on the technical corpus as well as the global vectors trained on the general corpus to represent the received CT's in 150 dimensional vectors. These vectors are then forwarded to the classifier to tell apart terms from non-terms. The highest-performing classifier is then found and used to initialize the system. We compare the results of our system with the results received from each of the term extraction tools used in isolation.

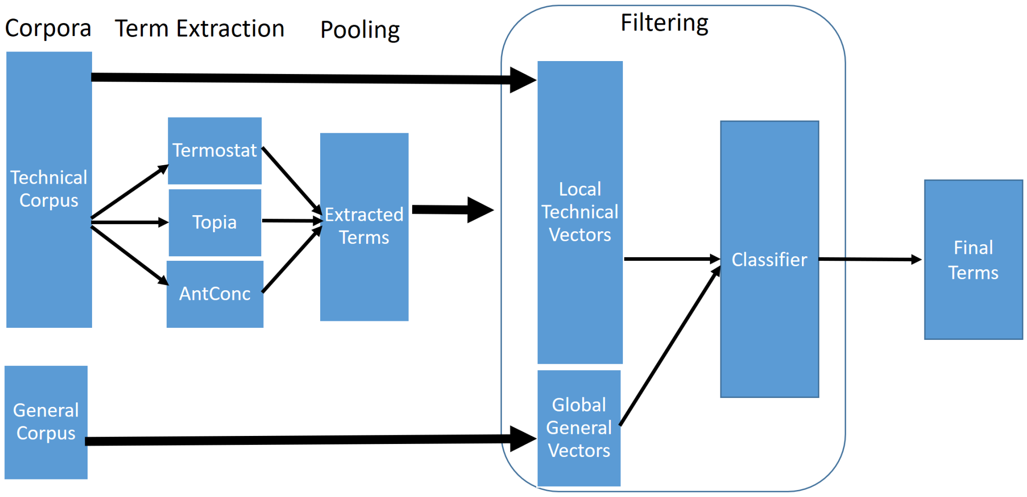

Figure 2: The figure depicts the overall system architecture of our method.

## 5 Annotation

To evaluate the performance of our system and compare it with the ATE tools used in isolation, two human annotators judged the terms extracted by the term extraction tools. The annotators used Term Evaluator[21] (Inkpen et al., 2016), a software program for annotating and evaluating terminology extraction, to judge the results. The annotators were asked to use their background knowledge of mathematics as the primary source of their judgment. In case of confusion, they could consult a mathematics dictionary of their choice. The annotations had kappa agreement scores of k = 0.70 for Topia, k = 0.84 for AntConc and, k = 0.53 for TermoStat. The annotation resulted in a dataset consisting of 954 instances with 325 positive and 629 negative cases, by which we assess the performance of the systems used in

[16] Another rule induction algorithm

[17] Available at: http://termostat.ling.umontreal.ca/

[18] topia.termextract 1.1.0 library available at: https://pypi.python.org/pypi/topia.termextract/

[19] Available at: http://www.laurenceanthony.net/software/antconc/

[20] AntConc has a keyword extraction module but no term extraction module.

[21] Available at: https://sourceforge.net/projects/termevaluator/

this study.

## 6 Experiments and Results

First, we aim to find the best-performing classifier(s), out of the ones tested, to be used in our system for each of the two architectures (CBOW and skip grams). We noticed that three classifiers, namely, SMO, logistic regression, and the multi-layer perceptron consistently outperformed the rest of the classifiers we examined (the full list is provided in section 4). JRip performed well, but its performance was consistently lower than the above-mentioned three classifiers. It should be noted, however, that we also have a greater dimension size for our vectors than Conrado et al. (2013), that is, 150 versus 19. Also the nature of the vectors is different in that they used feature vectors but we used embeddings. Nevertheless, we only show the results for these three classifiers. Figure 3 depicts the classifiers' performance with local-global vectors (LGVs) where the local vectors are trained with the CBOW architecture, and Figure 4 depicts the classifiers' performance with local-global vectors where the local vectors are trained with the skip-gram architecture. The classifiers' performance is presented as a function of the number of observed instances[22] (the amount of training data used), and the classifiers are tested on the rest of the instances (954 minus the number of observed instances). Instances are chosen randomly for training with a positive/negative ratio proportional to the dataset (i.e., $1/2$ respectively). All of the instances in the entire dataset are unique candidate terms. The performance is measured by F-measure in the figures. We compute only relative recall[23] throughout the experiments at this stage.

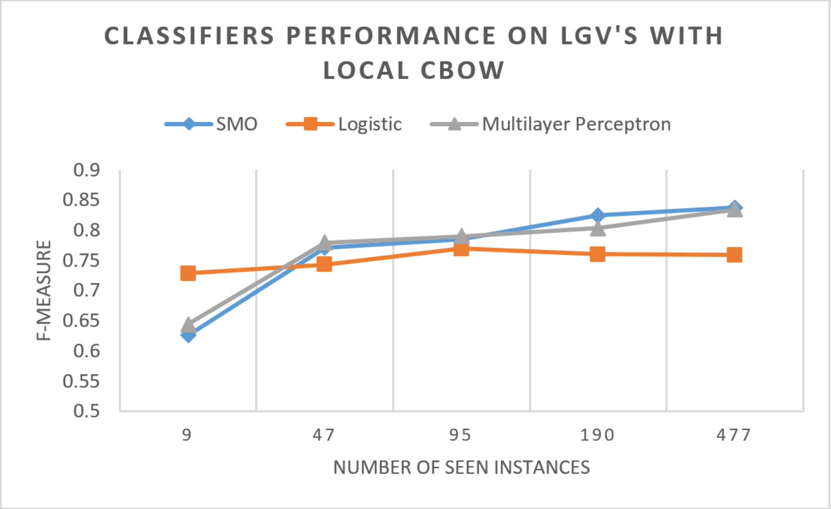

Figure 3: The figure displays the performance (F-measure) of the classifiers on local-global vectors with CBOW local vectors as a function of number of observed instances.

As shown in Figures 3 and 4 the CBOW architecture with the logistic regression classifier generalizes really well with as little data as only 9 instances. However, as soon as we add more instances, the multi-layer perceptron and SMO take the lead, outperforming one another in the process. However, the logistic regression classifier shows less improvement when subjected to more training data. Overall, we did not notice any considerable difference between the skip-gram and the CBOW architectures across the classifiers used for the purpose of unigram term extraction.

In practice, we prefer to show the system as little data as possible since extracting a few high-precision terms is relatively easy in real-world ATE; hence, we choose the local CBOW architecture with logistic

[22] The numbers shown on the $X$ axes of Figures 3 and 4 (i.e., 9, 47, 95, 190, and 477) are the results of splitting training and test data such that the training data is approximately 1%, 5%, 10%, 20%, and 50% of the entire dataset respectively. The most notable improvement is when we increase the training set from 9 to 47 and that is only 4% variation in the size of the test set but 10% improvement of performance on average for LGV's with local CBOW (Figure 3) and an average of 13% improvement of performance for LGV's with local skip-gram (Figure 4).

[23] The reason for resorting to relative recall is that having annotators go through the entire corpus to compute recall is time-consuming and labor-intensive at this phase of the project.

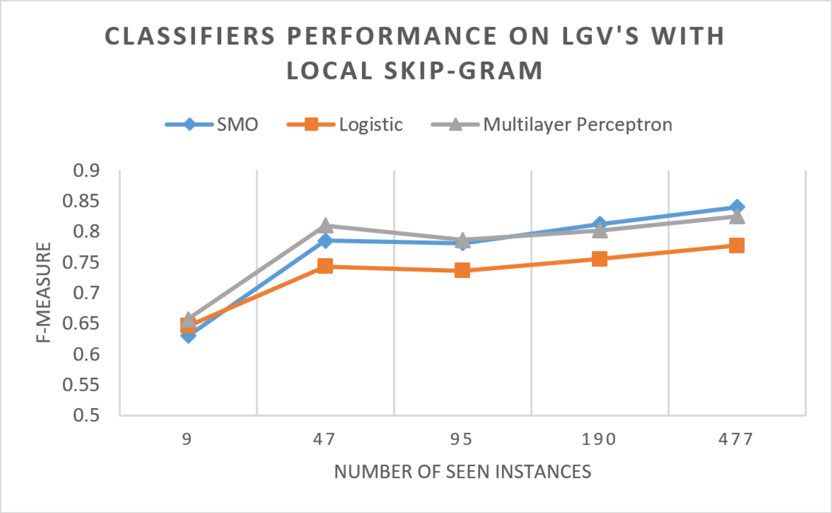

Figure 4: The figure displays the performance (F-measure) of the classifiers on local-global vectors with skip-gram local vectors as a function of number of observed instances.

regression classifier (trained on only 9 instances) as one configuration (our quickest learner), and the local skip-gram architecture coupled with multi-layer perceptron as the other configuration of our system (performs best among those trained on up to 47 instances) for the next experiment. We compare these two system configurations with a baseline and the initial term extraction tools, all tested on 907 (i.e., 954 - 47) remaining instances that are unseen to all of the systems under experiment. Table 1 compares the results of our system in unigram term extraction with individual term extraction tools and a frequency baseline that uses a stop-word filter (refer to section 4 for further details on the tools and the baseline). The results show that both of our system configurations achieve a substantial improvement over the other tools.

| Method | Precision | Recall | F-Measure |
|---|---|---|---|
| TermoStat | 0.371 | 0.528 | 0.436 |
| Improved Topia | 0.426 | 0.702 | 0.552 |
| Frequency + Stopword Filter | 0.407 | 0.514 | 0.454 |
| LGV9 (CBOW) + Logistic | 0.728 | 0.734 | 0.728 |
| LGV47 (skip-gram) + Multi-layer Perceptron | **0.809** | **0.811** | **0.810** |

Table 1: The table compares the results of two configurations of our system, LGV9 (using CBOW local vectors) and LGV47 (using skip-gram local vectors), with the term extraction tools used in isolation and with a frequency baseline.

## 7 Conclusion & Future Work

This paper offered a new ATE method that uses the distributed representation of words as a filter for the task of unigram term extraction. To do so, we leveraged the local-global embeddings to represent a term, its senses, and its behavior. The global word embeddings GloVe were pre-trained on general corpora, and we used the skip-gram and CBOW architectures to train local vectors on a technical domain corpus. This was done in order to preserve both the domain-specific and the general-domain information a word may possess, including its syntactic and semantic behavior. We showed that such a filter, with only as few instances as 9, can substantially improve the output of the three ATE tools in unigram term extraction. This indicates that with a) any high-precision (even with very low recall and F-measure) term extraction tool that outputs a few terms, b) a few random generic words in a language, and c) our filter, one can create a high-performance term extraction system for that language. Our method can also be used as a way to combine different tools to benefit from the advantages that each can offer, resulting in gain in performance. The use of the filter is not restricted to multiple term extraction tools and it can be applied as feasibly to any individual term extraction method. It is important to note that in our study the improvement in performance is not due to the merger of different tools but to a richer, more elaborate, and more informative representation of candidate terms. It was observed that the two local architectures, CBOW and skip-gram, do not show a considerable difference in capturing the technical sense and behavior of a word for unigram term extraction.

In future work, we plan to apply our local-global vectors directly to the corpus as a standalone term extraction tool. We also plan to extend the algorithm to detect multi-word terms in addition to unigram terms. It would be worthwhile to investigate if skip-gram and CBOW architectures can diverge in performance in extraction of terms that contain more than one word. Polysynthetic languages have a high morpheme-to-word ratio, that is, most of the grammatical and semantic information of a sentence is carried inside individual words, but continuous distributed models, including our LGV's, predominantly disregard word-internal structures. A very recent method based on the skip-gram architecture captures subword information in its word vectors (Bojanowski et al., 2016). We will address polysynthetic languages using enhanced LGV's as a next step. We further intend to compare our method with more available term extraction tools and methods. Applying our method to other domain corpora and datasets is another future direction for this research.

## References

L. Anthony. 2012. AntConc (Version 3.3.0) [Computer Software] http://www.laurenceanthony.net/.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Jürgen Broß and Heiko Ehrig. 2013. Terminology extraction approaches for product aspect detection in customer reviews. In Julia Hockenmaier and Sebastian Riedel, editors, *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 222–230. ACL.

T. Cabre-Castellvi, R. Estopa, and J. Vivaldi-Palatresi. 2001. Automatic term detection: A review of current systems. In D. Bourigault, C. Jacquemin, and M. LHomme, editors, *Recent Advances in Computational Terminology*. John Benjamins.

Teresa Mihwa Chung and Paul Nation. 2004. Identifying technical vocabulary. *System*, 32(2):251 – 263.

T. M. Chung. 2003. A corpus comparison approach for terminology extraction. *Terminology*, 9:221–246.

Merley Conrado, Thiago Pardo, and Solange Rezende. 2013. A machine learning approach to automatic term extraction using a rich feature set. page 16–23, Atlanta, Georgia, June. Association for Computational Linguistics.

P.W.D. Crippin, R. Donato, and D. Wright. 2007. *Calculus and Vectors*. Nelson Education Limited.

Patrick Drouin. 2003. Term extraction using non-technical corpora as a point of leverage. *Terminology*, 9(1):99–115.

Katerina T. Frantzi, Sophia Ananiadou, and Jun-ichi Tsujii. 1998. The c-value/nc-value method of automatic recognition for multi-word terms. In *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '98, pages 585–604, London, UK, UK. Springer-Verlag.

Diana Inkpen, T. Sima Paribakht, Farahnaz Faez, and Ehsan Amjadian. 2016. Term evaluator: A tool for terminology annotation and evaluation. *International Journal of Computational Linguistics and Applications*, pages –, December.

Azniah Ismail and Suresh Manandhar. 2010. Bilingual lexicon extraction from comparable corpora using in-domain terms. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 481–489, Stroudsburg, PA, USA. Association for Computational Linguistics.

K. Kageura and B. Umino. 1996. Methods of automatic term recognition: A review. *Terminology*, 3(2):259–289.

C. Kirkpatrick, B. Alldred, C. Chilvers, B. Farahani, K. Farentino, A. Lillo, I. Macpherson, J. Rodger, and S. Trew. 2007. *Nelson Advanced Functions*. Nelson Education.

Nikola Ljubešić, Špela Vintar, and Darja Fišer. 2012. Multi-word term extraction from comparable corpora by combining contextual and constituent clues. In Reinhard Rapp, Marko Tadić, Serge Sharoff, and Pierre Zweigenbaum, editors, *Proceedings of 5th Workshop on Building and Using Comparable Corpora (BUCC 2012)*, Istanbul, Turkey.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Ruslan Mitkov, Richard Evans, Constantin Orăsan, Iustin Dornescu, and Miguel Rios, 2012. *Text, Speech and Dialogue: 15th International Conference, TSD 2012, Brno, Czech Republic, September 3-7, 2012. Proceedings*, chapter Coreference Resolution: To What Extent Does It Help NLP Applications?, pages 16–27. Springer Berlin Heidelberg, Berlin, Heidelberg.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2265–2273. Curran Associates, Inc.

Rogelio Nazar and Maria Teresa Cabré. 2012. A machine learning approach to automatic term extraction using a rich feature set. page 209–217, Madrid, Spain.

Youngja Park, Roy J. Byrd, and Branimir K. Boguraev. 2002. Automatic Glossary Extraction: Beyond Terminology Identification. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

J. Platt. 1998. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado, June. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. `http://is.muni.cz/publication/884893/en`.

Anna ch Le Serrec, Marie-Claude L'Homme, Patrick Drouin, and Olivier Kraif. 2010. Automating the compilation of specialized dictionaries Use and analysis of term extraction and lexical alignment. *Terminology*, 16:77–107.

M. Small, C. Kirkpatrick, D. Zimmer, C. Chilvers, S. D Agostino, D. Duff, K. Farentino, I. Macpherson, J. Tonner, J. Williamson, and T. A. Yeager. 2005. *Principles of Mathematics 9*. Nelson Education Limited.

M. Small, C. Kirkpatrick, B. Alldred, S. Godin, A. Lillo, and A. Dmytriw. 2007a. *Functions 11*. Nelson Education Limited.

M. Small, C. Kirkpatrick, and A. Dmytriw. 2007b. *Functions and Applications 11*. Nelson Education Limited.

Peter D. Turney. 2000. Learning algorithms for keyphrase extraction. *Inf. Retr.*, 2(4):303–336, May.

Špela Vintar. 2010. Bilingual term recognition revisited: The bag-of-equivalents term alignment approach and its evaluation. *Terminology*, 16(2):141–158.

NICTA VRL. 2009. An unsupervised approach to domain-specific term extraction. In *Australasian Language Technology Association Workshop 2009*, page 94.

Thuy Vu, Ai Ti Aw, and Min Zhang. 2008. Term extraction through unithood and termhood unification. In *In Proc. of Intl Joint Conf on Natural Language Proc.*

Rui Wang, Wei Liu, and Chris McDonald. 2015. Corpus-independent generic keyphrase extraction using word embedding vectors. In *Deep Learning for Web Search and Data Mining*.

Yuhang Yang, Hao Yu, Yao Meng, Yingliang Lu, and Yingju Xia. 2010. Fault-tolerant learning for term extraction. In Ryo Otoguro, Kiyoshi Ishikawa, Hiroshi Umemoto, Kei Yoshimoto, and Yasunari Harada, editors, *PACLIC*, pages 321–330. Institute for Digital Enhancement of Cognitive Development, Waseda University.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. *CoRR*, abs/1605.07843.

Minoru Yoshida and Hiroshi Nakagawa, 2005. *Natural Language Processing – IJCNLP 2005: Second International Joint Conference, Jeju Island, Korea, October 11-13, 2005. Proceedings*, chapter Automatic Term Extraction Based on Perplexity of Compound Words, pages 269–279. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kalliopi Zervanou. 2010. Uvt: The uvt term extraction system in the keyphrase extraction task. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 194–197, Uppsala, Sweden, July. Association for Computational Linguistics.