Leveraging distributed representations and lexico-syntactic fixedness for token-level prediction of the idiomaticity of English verb–noun combinations

Milton King and **Paul Cook**
Faculty of Computer Science, University of New Brunswick
Fredericton, NB E3B 5A3 Canada
milton.king@unb.ca, paul.cook@unb.ca

Abstract

Verb–noun combinations (VNCs) — e.g., *blow the whistle*, *hit the roof*, and *see stars* — are a common type of English idiom that are ambiguous with literal usages. In this paper we propose and evaluate models for classifying VNC usages as idiomatic or literal, based on a variety of approaches to forming distributed representations. Our results show that a model based on averaging word embeddings performs on par with, or better than, a previously-proposed approach based on skip-thoughts. Idiomatic usages of VNCs are known to exhibit lexico-syntactic fixedness. We further incorporate this information into our models, demonstrating that this rich linguistic knowledge is complementary to the information carried by distributed representations.

1 Introduction

Multiword expressions (MWEs) are combinations of multiple words that exhibit some degree of idiomaticity (Baldwin and Kim, 2010). Verb–noun combinations (VNCs), consisting of a verb with a noun in its direct object position, are a common type of semantically-idiomatic MWE in English and cross-lingually (Fazly et al., 2009). Many VNCs are ambiguous between MWEs and literal combinations, as in the following examples of *see stars*, in which 1 is an idiomatic usage (i.e., an MWE), while 2 is a literal combination.[1]

1. Hereford United were seeing stars at Gillingham after letting in 2 early goals

2. Look into the night sky to see the stars

[1]These examples, and idiomaticity judgements, are taken from the VNC-Tokens dataset (Cook et al., 2008).

MWE identification is the task of automatically determining which word combinations at the token-level form MWEs (Baldwin and Kim, 2010), and must be able to make such distinctions. This is particularly important for applications such as machine translation (Sag et al., 2002), where the appropriate meaning of word combinations in context must be preserved for accurate translation.

In this paper, following prior work (e.g., Salton et al., 2016), we frame token-level identification of VNCs as a supervised binary classification problem, i.e., idiomatic vs. literal. We consider a range of approaches to forming distributed representations of the context in which a VNC occurs, including word embeddings (Mikolov et al., 2013), word embeddings tailored to representing sentences (Kenter et al., 2016), and skip-thoughts sentence embeddings (Kiros et al., 2015). We then train a support vector machine (SVM) on these representations to classify unseen VNC instances. Surprisingly, we find that an approach based on representing sentences as the average of their word embeddings performs comparably to, or better than, the skip-thoughts based approach previously proposed by Salton et al. (2016).

VNCs exhibit lexico-syntactic fixedness. For example, the idiomatic interpretation in example 1 above is typically only accessible when the verb *see* has active voice, the determiner is null, and the noun *star* is in plural form, as in *see stars* or *seeing stars*. Usages with a determiner (as in example 2), a singular noun (e.g., *see a star*), or passive voice (e.g., *stars were seen*) typically only have the literal interpretation.

In this paper we further incorporate knowledge of the lexico-syntactic fixedness of VNCs — automatically acquired from corpora using the method of Fazly et al. (2009) — into our various embedding-based approaches. Our experimental results show that this leads to substantial improve-

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 345–350
Melbourne, Australia, July 15 - 20, 2018.

ments, indicating that this rich linguistic knowledge is complementary to that available in distributed representations.

2 Related work

Much research on MWE identification has focused on specific kinds of MWEs (e.g., Patrick and Fletcher, 2005; Uchiyama et al., 2005), including English VNCs (e.g., Fazly et al., 2009; Salton et al., 2016), although some recent work has considered the identification of a broad range of kinds of MWEs (e.g., Schneider et al., 2014; Brooke et al., 2014; Savary et al., 2017).

Work on MWE identification has leveraged rich linguistic knowledge of the constructions under consideration (e.g., Fazly et al., 2009; Fothergill and Baldwin, 2012), treated literal and idiomatic as two senses of an expression and applied approaches similar to word-sense disambiguation (e.g., Birke and Sarkar, 2006; Hashimoto and Kawahara, 2008), incorporated topic models (e.g., Li et al., 2010), and made use of distributed representations of words (Gharbieh et al., 2016).

In the most closely related work to ours, Salton et al. (2016) represent token instances of VNCs by embedding the sentence that they occur in using skip-thoughts (Kiros et al., 2015) — an encoder–decoder model that can be viewed as a sentence-level counterpart to the word2vec (Mikolov et al., 2013) skip-gram model. During training the target sentence is encoded using a recurrent neural network, and is used to predict the previous and next sentences. Salton et al. then use these sentence embeddings, representing VNC token instances, as features in a supervised classifier. We treat this skip-thoughts based approach as a strong baseline to compare against.

Fazly et al. (2009) formed a set of eleven lexico-syntactic patterns for VNC instances capturing the voice of the verb (active or passive), determiner (e.g., *a*, *the*), and number of the noun (singular or plural). They then determine the canonical form, $C(v,n)$, for a given VNC as follows:[2]

$$C(v,n) = \{pt_k \in P | z(v,n,pt_k) > T_z\} \quad (1)$$

where P is the set of patterns, T_z is a predetermined threshold, which is set to 1, and $z(v,n,pt_k)$ is calculated as follows:

$$z(v,n,pt_k) = \frac{f(v,n,pt_k) - \overline{f}}{s} \quad (2)$$

where $f(\cdot)$ is the frequency of a VNC occurring in a given pattern in a corpus,[3] and $\overline{f}$ and s are the mean and standard deviations for all patterns for the given VNC, respectively.

Fazly et al. (2009) showed that idiomatic usages of a VNC tend to occur in that expression's canonical form, while literal usages do not. This approach provides a strong, linguistically-informed, unsupervised baseline, referred to as CForm, for predicting whether VNC instances are idiomatic or literal. In this paper we incorporate knowledge of canonical forms into embedding-based approaches to VNC token classification, and show that this linguistic knowledge can be leveraged to improve such approaches.

3 Models

We describe the models used to represent VNC token instances below. For each model, a linear SVM classifier is trained on these representations.

3.1 Word2vec

We trained word2vec's skip-gram model (Mikolov et al., 2013) on a snapshot of Wikipedia from September 2015, which consists of approximately 2.6 billion tokens. We used a window size of ± 8 and 300 dimensions. We ignore all words that occur less than fifteen times in the training corpus, and did not set a maximum vocabulary size. We perform negative sampling and set the number of training epochs to five. We used batch processing with approximately $10k$ words in each batch.

To embed a given a sentence containing a VNC token instance, we average the word embeddings for each word in the sentence, including stopwords.[4] Prior to averaging, we normalize each embedding to have unit length.

3.2 Siamese CBOW

The Siamese CBOW model (Kenter et al., 2016) learns word embeddings that are better able to represent a sentence through averaging than conventional word embeddings such as skip-gram or CBOW. We use a Siamese CBOW model that was pretrained on a snapshot of Wikipedia from November 2012 using randomly initialized word

[2] In a small number of cases a VNC is found to have a small number of canonical forms, as opposed to just one.

[3] Fazly et al. (2009) used the British National Corpus (Burnard, 2000).

[4] Preliminary experiments showed that models performed better when stopword removal was not applied.

embeddings.[5] Similarly to the word2vec model, to embed a given sentence containing a VNC instance, we average the word embeddings for each word in the sentence.

3.3 Skip-thoughts

We use a publicly-available skip-thoughts model, that was pre-trained on a corpus of books.[6] We represent a given sentence containing a VNC instance using the skip-thoughts encoder. Note that this approach is our re-implementation of the skip-thoughts based method of Salton et al. (2016), and we use it as a strong baseline for comparison.

4 Data and evaluation

In this section, we discuss the dataset used in our experiments, and the evaluation of our models.

4.1 Dataset

We use the VNC-Tokens dataset (Cook et al., 2008) — the same dataset used by Fazly et al. (2009) and Salton et al. (2016) — to train and evaluate our models. This dataset consists of sentences containing VNC usages drawn from the British National Corpus (Burnard, 2000),[7] along with a label indicating whether the VNC is an idiomatic or literal usage (or whether this cannot be determined, in which case it is labelled "unknown").

VNC-Tokens is divided into DEV and TEST sets that each include fourteen VNC types and a total of roughly six hundred instances of these types annotated as literal or idiomatic. Following Salton et al. (2016), we use DEV and TEST, and ignore all token instances annotated as "unknown".

Fazly et al. (2009) and Salton et al. (2016) structured their experiments differently. Fazly et al. report results over DEV and TEST separately. In this setup TEST consists of expressions that were not seen during model development (done on DEV). Salton et al., on the other hand, merge DEV and TEST, and create new training and testing sets, such that each expression is present in the training and testing data, and the ratio of idiomatic to literal usages of each expression in the training data is roughly equal to that in the testing data.

We borrowed ideas from both of these approaches in structuring our experiments. We retain the type-level division of Fazly et al. (2009) into DEV and TEST. We then divide each of these into training and testing sets, using the same ratios of idiomatic to literal usages for each expression as Salton et al. (2016). This allows us to develop and tune a model on DEV, and then determine whether, when retrained on instances of unseen VNCs in (the training portion of) TEST, that model is able to generalize to new VNCs without further tuning to the specific expressions in TEST.

[5] https://bitbucket.org/TomKenter/siamese-cbow

[6] https://github.com/ryankiros/skip-thoughts

[7] http://www.natcorp.ox.ac.uk

Model	Penalty cost				
	0.01	0.1	1	10	100
Word2vec	0.619	0.654	0.818	**0.830**	0.807
Siamese CBOW	0.619	0.621	0.665	0.729	**0.763**
Skip-thoughts	0.661	0.784	**0.803**	0.800	0.798

Table 1: Accuracy on DEV while tuning the penalty cost for the SVM for each model. The highest accuracy for each model is shown in boldface.

4.2 Evaluation

The proportion of idiomatic usages in the testing portions of both DEV and TEST is 63%. We therefore use accuracy to evaluate our models following Fazly et al. (2009) because the classes are roughly balanced. We randomly divide both DEV and TEST into training and testing portions ten times, following Salton et al. (2016). For each of the ten runs, we compute the accuracy for each expression, and then compute the average accuracy over the expressions. We then report the average accuracy over the ten runs.

5 Experimental results

In this section we first consider the effect of tuning the cost parameter of the SVM for each model on DEV, and then report results on DEV and TEST using the tuned models.

5.1 Parameter tuning

We tune the SVM for each model on DEV by carrying out a linear search for the penalty cost from 0.01–100, increasing by a factor of ten each time. Results for this parameter tuning are shown in Table 1. These results highlight the importance of choosing an appropriate setting for the penalty cost. For example, the accuracy of the word2vec model ranges from 0.619–0.830 depending on the cost setting. In subsequent experiments, for each

Model	DEV		TEST	
	−CF	+CF	−CF	+CF
CForm	-	0.721	-	0.749
Word2vec	**0.830**	**0.854**	**0.804**	**0.852**
Siamese CBOW	0.763	0.774	0.717	0.779
Skip-thoughts	0.803	0.827	0.786	0.842

Table 2: Accuracy on DEV and TEST for each model, without (−CF) and with (+CF) the canonical form feature. The highest accuracy for each setting on each dataset is shown in boldface.

model, we use the penalty cost that achieves the highest accuracy in Table 1.

5.2 DEV and TEST results

In Table 2 we report results on DEV and TEST for each model, as well as the unsupervised CForm model of Fazly et al. (2009), which simply labels a VNC as idiomatic if it occurs in its canonical form, and as literal otherwise. We further consider each model (other than CForm) in two setups. −CF corresponds to the models as described in Section 3. +CF further incorporates lexico-syntactic knowledge of canonical forms into each model by concatenating the embedding representing each VNC token instance with a one-dimensional vector which is one if the VNC occurs in its canonical form, and zero otherwise.

We first consider results for the −CF setup. On both DEV and TEST, the accuracy achieved by each supervised model is higher than that of the unsupervised CForm approach, except for Siamese CBOW on TEST. The word2vec model achieves the highest accuracy on DEV and TEST of 0.830 and 0.804, respectively. The difference between the word2vec model and the next-best model, skip-thoughts, is significant using a bootstrap test (Berg-Kirkpatrick et al., 2012) with $10k$ repetitions for DEV ($p = 0.006$), but not for TEST ($p = 0.051$). Nevertheless, it is remarkable that the relatively simple approach to averaging word embeddings used by word2vec performs as well as, or better than, the much more complex skip-thoughts model used by Salton et al. (2016).[8]

Turning to the +CF setup, we observe that, for both DEV and TEST, each model achieves higher accuracy than in the −CF setup.[9] All of these differences are significant using a bootstrap test ($p < 0.002$ in each case). In addition, each method outperforms the unsupervised CForm approach on both DEV and TEST. These findings demonstrate that the linguistically-motivated, lexico-syntactic knowledge encoded by the canonical form feature is complementary to the information from a wide range of types of distributed representations. In the +CF setup, the word2vec model again achieves the highest accuracy on both DEV and TEST of 0.854 and 0.852, respectively.[10] The difference between the word2vec model and the next-best model, again skip-thoughts, is significant for both DEV and TEST using a bootstrap test ($p < 0.05$ in each case).

To better understand the impact of the canonical form feature when combined with the word2vec model, we compute the average precision, recall, and F1 score for each MWE for both the positive (idiomatic) and negative (literal) classes, for each run on TEST.[11] For a given run, we then compute the average precision, recall, and F1 score across all MWEs, and then the average over all ten runs. We do this using CForm, and the word2vec model with and without the canonical form feature. Results are shown in Table 3. In line with the findings of Fazly et al. (2009), CForm achieves higher precision and recall on idiomatic usages than literal ones. In particular, the relatively low recall for the literal class indicates that many literal usages occur in a canonical form. Comparing the word2vec model with and without the canonical form feature, we see that, when this feature is used, there is a relatively larger increase in precision and recall (and F1 score) for the literal class, than for the idiomatic class. This indicates that, although the

[8]The word2vec and skip-thoughts models were trained on different corpora, which could contribute to the differences in results for these models. We therefore carried out an additional experiment in which we trained word2vec on BookCorpus, the corpus on which skip-thoughts was trained. This new word2vec model achieved accuracies of 0.825 and 0.809, on DEV and TEST, respectively, which are also higher accuracies than those obtained by the skip-thoughts model.

[9]In order to determine that this improvement is due to the information about canonical forms carried by the additional feature in the +CF setup, and not due to the increase in number of dimensions, we performed additional experiments in which we concatenated the embedding representations with a random binary feature, and with a randomly chosen value between 0 and 1. For each model, neither of these approaches outperformed that model using the +CF setup.

[10]In the +CF setup, the word2vec model using embeddings that were trained on the same corpus as skip-thoughts achieved accuracies of 0.846 and 0.851, on DEV and TEST, respectively. These are again higher accuracies than the corresponding setup for the skip-thoughts model.

[11]We carried out the same analysis on DEV. The findings were similar.

Model	Idiomatic			Literal			Ave. F
	P	R	F	P	R	F	
CForm	0.766	0.901	0.794	0.668	0.587	0.576	0.685
Word2vec −CF	0.815	0.879	0.830	0.627	0.542	0.556	0.693
Word2vec +CF	0.830	0.892	0.848	0.758	0.676	0.691	0.770

Table 3: Precision (P), recall (R), and F1 score (F), for the idiomatic and literal classes, as well as average F1 score (Ave. F), for TEST.

canonical form feature itself performs relatively poorly on literal usages, it provides information that enables the word2vec model to better identify literal usages.

6 Conclusions

Determining whether a usage of a VNC is idiomatic or literal is important for applications such as machine translation, where it is vital to preserve the meanings of word combinations. In this paper we proposed two approaches to the task of classifying VNC token instances as idiomatic or literal based on word2vec embeddings and Siamese CBOW. We compared these approaches against a linguistically-informed unsupervised baseline, and a model based on skip-thoughts previously applied to this task (Salton et al., 2016). Our experimental results show that a comparatively simple approach based on averaging word embeddings performs at least as well as, or better than, the approach based on skip-thoughts. We further proposed methods to combine linguistic knowledge of the lexico-syntactic fixedness of VNCs — so-called "canonical forms", which can be automatically acquired from corpora via statistical methods — with the embedding based approaches. Our findings indicate that this rich linguistic knowledge is complementary to that available in distributed representations.

Alternative approaches to embedding sentences containing VNC instances could also be considered, for example, FastSent (Hill et al., 2016). However, all of the models we used represent the context of a VNC by the sentence in which it occurs. In future work we therefore also intend to consider approaches such as context2vec (Melamud et al., 2016) which explicitly encode the context in which a token occurs. Finally, one known challenge of VNC token classification is to develop models that are able to generalize to VNC types that were not seen during training (Gharbieh et al., 2016). In future work we plan to explore this experimental setup.

References

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing*, CRC Press, Boca Raton, USA. 2nd edition.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, pages 995–1005. http://www.aclweb.org/anthology/D12-1091.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*. Trento, Italy, pages 329–336. http://aclweb.org/anthology/E/E06/E06-1042.pdf.

Julian Brooke, Vivian Tsang, Graeme Hirst, and Fraser Shein. 2014. Unsupervised multiword segmentation of large corpora using prediction-driven decomposition of n-grams. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland, pages 753–761.

Lou Burnard. 2000. *The British National Corpus Users Reference Guide*. Oxford University Computing Services.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. 2008. The VNC-Tokens Dataset. In *Proceedings of the LREC Workshop on Towards a Shared Task for Multiword Expressions (MWE 2008)*. Marrakech, Morocco, pages 19–22. http://www.lrec-conf.org/proceedings/lrec2008/workshops/W20_Proceedings.pdf.

Afsaneh Fazly, Paul Cook, and Suzanne Stevenson. 2009. Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics* 35(1):61–103. http://dx.doi.org/10.1162/coli.08-010-R1-07-048.

Richard Fothergill and Timothy Baldwin. 2012. Combining resources for mwe-token classification. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada, pages 100–104. http://www.aclweb.org/anthology/S12-1017.

Waseem Gharbieh, Virendra C Bhavsar, and Paul Cook. 2016. A word embedding approach to identifying verb–noun idiomatic combinations. In *Proceedings of the 12th Workshop on Multiword Expressions (MWE 2016)*. Association for Computational Linguistics, Berlin, Germany, pages 112–118. http://aclweb.org/anthology/W/W16/W16-1817.pdf.

Chikara Hashimoto and Daisuke Kawahara. 2008. Construction of an idiom corpus and its application to idiom identification based on WSD incorporating idiom-specific features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii, pages 992–1001.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 1367–1377. http://www.aclweb.org/anthology/N16-1162.

Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese cbow: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 941–951. http://www.aclweb.org/anthology/P16-1089.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., pages 3276–3284.

Linlin Li, Benjamin Roth, and Caroline Sporleder. 2010. Topic models for word sense disambiguation and token-based idiom detection. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden, pages 1138–1147.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. pages 51–61. http://www.aclweb.org/anthology/K16-1006.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at the International Conference on Learning Representations, 2013*. Scottsdale, USA. https://arxiv.org/abs/1301.3781.

Jon Patrick and Jeremy Fletcher. 2005. Classifying verb-particle constructions by verb arguments. In *Proceedings of the Second ACL-SIGSEM Workshop on the Linguistic Dimensions of Prepositions and their use in Computational Linguistics Formalisms and Applications*. Colchester, UK, pages 200–209.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2002)*. pages 1–15. http://dl.acm.org/citation.cfm?id=647344.724004.

Giancarlo Salton, Robert Ross, and John Kelleher. 2016. Idiom token classification using sentential distributed semantics. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 194–204. http://www.aclweb.org/anthology/P16-1019.

Agata Savary, Carlos Ramisch, Silvio Cordeiro, Federico Sangati, Veronika Vincze, Behrang QasemiZadeh, Marie Candito, Fabienne Cap, Voula Giouli, Ivelina Stoyanova, and Antoine Doucet. 2017. The parseme shared task on automatic identification of verbal multiword expressions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*. Association for Computational Linguistics, Valencia, Spain, pages 31–47. http://www.aclweb.org/anthology/W17-1704.

Nathan Schneider, Emily Danchik, Chris Dyer, and Noah A. Smith. 2014. Discriminative lexical semantic segmentation with gaps: Running the mwe gamut. *Transactions of the Association of Computational Linguistics* 2:193–206.

Kiyoko Uchiyama, Timothy Baldwin, and Shun Ishizaki. 2005. Disambiguating Japanese compound verbs. *Computer Speech and Language, Special Issue on Multiword Expressions* 19(4):497–512.