A Survey of Word Embedding Literature: Context Representations and the Challenge of Ambiguity

Article · April 2016

1 author:

Jordy Van Landeghem
University of Leuven
7 PUBLICATIONS **0** CITATIONS

SEE PROFILE

KU LEUVEN

FACULTY OF ENGINEERING SCIENCE

Department of Computer Science

A Survey of Word Embedding Literature

Context-vector representations and the challenge of ambiguity

Jordy Van Landeghem

Master Artificial Intelligence
Speech and Language Technology

april 2016

1. Introduction

Understanding the semantics of a word and in extension that of larger units (phrase, sentence, paragraph, document, collection) is the core business of respectively Natural Language Processing (NLP) and Text-Based Information Retrieval ([TB]IR). In order to teach machines to gain a deep understanding of words (and concepts), we need to define a representation which they can operate on, a so-called "word representation". The choice of how to represent words in such a way that as much information as possible is encoded in them poses a fundamental challenge for the NLP & IR community. Recent advances in Machine Learning have made it possible to train more complex models on much larger datasets, thereby outperforming simpler models. *Neural* or *Word Embeddings* have gained universal recognition thanks to the recent work of Mikolov et al. (2013a, b), in which the presented log-linear models, *continuous Bag-of-Words Model* (CBOW) and *continuous Skip-gram Model* (SG), allow for very efficient estimation of continuous-space word representations from huge datasets (e.g. 1.6 billion words Mikolov 2013a). Current research into word representations has been dominated by this new generation of word embeddings. Their large success and extreme popularity can also be attributed to the open-source distribution of the word2vec software package (Mikolov et al. 2013), which contains pre-computed embeddings and production code, and amongst others GloVe (Pennington et al. 2014) and Deeplearning4j (Gibson 2015). Recent contributions have highlighted the importance of these neural embeddings for NLP and the effect of their quality in experimental studies.

This report will offer a critical survey of reported advances in word embedding literature. In the following sections, we will focus on specific questions concerning the induction of unimodal embeddings. Firstly, we will zoom in on approaches that take on the challenge of ambiguity in vector space. Secondly, a discussion is started to compare the performance of traditional count-based models and the new neural-network-based predictive methods on a variety of NLP tasks. We will briefly touch upon the current state-of-the-art research and review the possibly new insights that have been generated. The structure of the report will be as follows: Each section will start off with a description of the problem, followed by suggested solutions, which will be illustrated with exemplary papers. Finally, all insights will be bundled and motivated in a critical discussion. At the end of each discussion, some future paths will be suggested for word embedding research in general.

2. Vector representations vs. ambiguity

The problem of ambiguity is one of the most difficult challenges for the processing of natural language in texts. Research into this pervasive, yet essential phenomenon in language has a large tradition in many disciplines (cognitive science, philosophy to name just a few), but NLP is the foremost to try and tackle this obstacle head-on, because many

of its applications depend on the correct usage of words. Ambiguity presents itself on all levels of language: if it is on the word-level, such as "sink", which can be a verb or a noun, it is called lexical ambiguity. If else it occurs on the clause or sentence-level, it is referred to as structural ambiguity[1]. Ambiguity in language has received special attention in the field of NLP, since its technological aim is to enable computers to be used as aids in the analysis and processing of natural language. However, computers do not possess any knowledge of the world or enough context to disambiguate natural sentences like the following:

(1) Labour grills the British Prime Minister over the leaked Panama Papers.

By purely relying on semantics, this highly-ambiguous sentence can be interpreted in the following ways:

Interpretation 1:

"The British Prime Minister" is being questioned and criticized by "Labour" about his involvement in "leaked Panama Papers".

Interpretation 2:

"The British Prime Minister" is being cooked alive over a fire lit up with "leaked Panama Papers" by "Labour".

Humans typically resolve these type of ambiguous sentences with context and world knowledge. Even if one has never heard of the recent offshore trust scandal, appropriately dubbed "Panama Papers" ; the British, centre-left political "Labour" party ; or who the current British Prime Minister is, no human would ever consider the second interpretation to be very probable or even possible, although the lexical semantics of "grill" allows both interpretations. Computational systems lack this human ability and therefore intelligent approaches will need to be developed. The NLP community has proposed two potential paths in the form of knowledge-based and statistical systems. Vector representations for words belong loosely to the latter type in which large quantities of language data combined with statistical techniques aim to model language correctly and sufficiently well. However, word embeddings do not account for ambiguity. For the remainder of this section, we will review recent word embedding literature for approaches that have been developed to deal with the problem of ambiguity.

2.1. Ambiguity in embeddings

Word embeddings as presented by Mikolov et al. (2013a, b) form an interesting mode of representation and have been shown to perform well in a variety of NLP and IR tasks. The recent popularity of embeddings stems from the observation that they "are able to capture surprisingly nuanced semantics, even in absence of sentence structure (Chen et al. 2013: 2)".

[1] (see *http://www.csi.uottawa.ca/tanka/files/complexities.html* for an exhaustive list of ambiguity problems in language and Jurafsky and Martin (2008) for examples and grammatical discussion of structural ambiguity)

Words and their windowed (and thus local) context information are effectively represented in vector space, which allows for geometry-based similarity comparisons. However, vector representations such as embeddings are still faced by the hard challenge of representing words that have multiple meanings or senses. A point of criticism that can be rightfully uttered against the type of embeddings in Mikolov et al. (2013) is that words are captured in a single vector representation, which does not account for the possible polysemy or homonymy of the represented words.[2]

Below we will illustrate by means of a model trained with the word2vec package on the Text8 corpus (cf. exercise session TBIR 2016) how and where ambiguity is pervasive in embeddings. After computing semantic similarity scores for the known-to-be ambiguous words *apple*, *apples* and *fluke*, the resulting ranked lists will be qualitatively analysed.

apple	position:	1221	**apples**	position:	12247	**fluke**	position:	28287	
#	Word	Cosine dist.	#	Word	Cosine dist.	#	Word	Cosine dist.	Sense
1	macintosh	0.89	1	pears	0.85	1	anchors	0.78	1
2	performa	0.81	2	raspberry	0.82	2	anchors	0.72	1
3	hypercard	0.78	3	raisins	0.81	3	protruding	0.72	2
4	desktop	0.75	4	plums	0.81	4	flukes	0.69	/
5	amiga	0.75	5	fruits	0.81	5	rudder	0.69	1
6	iigs	0.74	6	cherries	0.81	6	bulbous	0.68	1, 2
7	imac	0.73	7	beans	0.8	7	tapered	0.67	2, 3, 4, 5
8	atari	0.72	8	apricots	0.8	8	hook	0.66	1, 2
9	compaq	0.72	9	sauerkraut	0.8	9	waterproof	0.66	1
10	quickdraw	0.72	10	tomatoes	0.8	10	hulled	0.65	1
11	amigaone	0.71	11	lentils	0.8	11	shells	0.65	1
12	microsoft	0.71	12	caramel	0.8	12	aft	0.65	1
13	macs	0.71	13	cabbage	0.79	13	setae	0.65	3
14	mac	0.71	14	berries	0.79	14	clipped	0.64	2
15	claris	0.7	15	watermelon	0.78	15	sharpening	0.64	2
16	ibm	0.7	16	currants	0.77	16	screw	0.64	2
17	raskin	0.7	17	artichokes	0.77	17	grooves	0.63	2
18	iix	0.7	18	salads	0.77	18	clutching	0.63	1
19	pc	0.69	19	peas	0.76	19	riveted	0.63	2
20	workstation	0.69	20	fruit	0.76	20	fins	0.63	4, 5

Table 1: ranked list with 20 most similar context words for *apple*, *apples* and *fluke*.

The resulting ranked list for the word *apple* shows that the embedding is biased towards the technological domain, which is probably caused by "Apple" in its corporate name sense. The

[2] "Other embeddings also do not address this inconsistency explicitly. In the words of Chen et al. (2013) "not all embeddings are created equal". However, for the sake of illustration we will evaluate the word2vec embeddings used in the exercise sessions(TBIR 2016).

type of ambiguity that is evident from this embedding could be appropriately dubbed a 'capitonym' (Lederer 1998).

However, the observed similarity scores for *apples* show high similarity with its singular vector variant and do prefer the fruit denotation. Next to that, we also observe the high similarity with words denoting vegetables, which might be explained by their attributional similarity (edible, healthy, food source, large color variety…).

The word *fluke* can certainly be considered a highly ambiguous word, seeing that it can denote a large variety of meanings which often are not even known by all language users and might rely on the user's knowledge of the world. A quick survey of WordNet and the Oxford English Dictionary reveals the following varying, but often interrelated meanings:

FLUKE

#	Part-of-Speech	Sense	General context
1	Noun	Bladed end of an anchor	boats
2	Noun	Barbed edge of an arrow/harpoon	general form
3	Noun	Type of parasitic flatworm	animal
4	Noun	Type of flatfish, flounder	animal
5	Noun	Fins on a whale's tail	animal anatomy
6	Noun	Stroke of luck	chance
7	Verb	Achieve by luck rather than skill	chance

Table 2: a short survey of the meanings in usage for the word fluke

On the basis of the ranked list and a subjective evaluation, it has been observed that context words belonging to the first two senses (both are equally related to half of the words in the ranked list) are overly present in the induced embedding. The next three senses related to 'animals' are only related to a small percentage (+- 10%) of highly similar words, while the 'chance' context is not even present in the highest similar scored words. One important observation has to be made here: It is clear that the resulting embeddings are dependent on the data on which they have been trained. It should be noted that only a small corpus has been used and thus not all senses have been captured (or cannot be deduced from the ranked lists). From the previous observation it is obvious why there had been such a demand for efficient techniques to induce embeddings from very large datasets (until Mikolov et al. 2013 introduced their log-linear models). Nevertheless, for illustrative purposes a small corpus suffices to identify already some linguistic regularities. However, these unsupervised lexical embeddings do not distinguish between word senses and collect all usage (viz. context) cases in 1 single vector representation. In the following section, we will review innovative embedding approaches and how they tackle ambiguity.

2.2. Suggested solutions

Recent research in embeddings has focused mostly on tuning embeddings to various tasks with the help of extra information. Generally, the various 'new' embedding techniques

show good or increased accuracy when evaluated on a range of classic NLP tasks, which in consequence might imply good handling of ambiguity. For example, "Two/too simple adaptations" of word2vec presented by Ling et al. (2015) seek to improve the quality of embeddings for syntactically-motivated tasks. They have claimed that earlier embedding models perform "sub-optimal" for (mainstream) syntax-based tasks such as POS-tagging and dependency parsing, which are known to suffer from ambiguity (Jurafsky and Martin 2008: ch. 5, 13, 14). By introduction of a small modification to the original word2vec models, viz. to make them aware of the relative positioning in which context words occur, consequentially dubbed (unsupervised) *Structured Skip-Gram* and *Continuous Window* , Ling et al. (2015) have shown that the proposed models (Wang2Vec) slightly outperform original embedding methods and can generalize even under noisy conditions. Nevertheless, we should discern that including word order information only delivers small-margin improvements for parsing and dependency albeit with only a small extra cost of computation speed. Whereas the previous proposed techniques only adapt the neural network architectures to account for the ordering of context words with the goal of improving the learned embeddings, other approaches exploit extra factors (or features) from supervised data to tailor embeddings for the intended tasks. Seeing that humans need "context" or "(world) knowledge" to disambiguate natural language, computers arguably do so as well and therefore should be provided with them. The main idea behind the following research papers is that unsupervised vectors do not distinguish between word senses and are not able to capture all aspects of language structure, so structural features ought to be added for better performance and additionally, combined objective methods can provide even further improvement.

Yu and Dredze (2014) argue that embeddings produce a reasonable performance boost when incorporated together with prior knowledge contained in semantic resources [demonstrated with the Paraphrase Database (Ganitkevitch et al. 2013) and WordNet (Fellbaum, 1999)] into established NLP pipelines. Levy and Goldberg (2014) experiment with syntactic contexts derived from automatically produced dependency parse-trees to create *Dependency Embeddings*. They propose a generalisation of the word2vec models that combines linear context, which captures mostly broad topical similarities, with indirect dependency-based context, capturing functional similarities. Cross et al. (2015) expound on the idea of leveraging dependency relationships between words, but add sentence structure to improve the semantics that embeddings capture. This has proven to be helpful for phrase-oriented tasks (such as semantic role labelling, frame labelling …) that rely on the context of units larger than words. Moreover, in their exploration of joint learning of embeddings and sequential training from sentence to lower levels they conclude that combined vectors are more versatile for different tasks and are able to capture a fine-grained interplay between topical and functional similarity. *Factor-based Compositional Embedding Models* (Yu and Dredze 2015, Gormley, Yu, Dredze 2015) seek to build a representation for even larger structures based on their component embeddings, which sum up information about words

and their interactions. With the aid of a small set of sentence annotation (POS tags, dependency parses and named entities) these innovative models have been demonstrated to outperform the state-of-the-art SVM (Semeval 2010), which drew upon a super-rich feature set. Another novel embedding approach stands out between the above in that it succeeds in capturing the semantic variability of words in a dynamic representation without explicit definition (thesauri or dictionaries), no pre-processing (features or factors) and/or ad-hoc clustering (cf. infra). *Infinite Dimensional Skip-Gram* (Nalisnick & Ravi 2015) is a not linguistically-inspired technique that allows vectors of specific words to grow naturally based on how well they can predict their context, thus efficiently shortening or elongating vectors for respectively simple or complex (i.e. vague) words.

2.3. Word-Sense Disambiguation of embeddings

All the above approaches have led to good advancements on a wide range of tasks, yet the various embedding technique share a common representational problem: even though the resulting embeddings have been fine-tuned with features during training to make-up for deficiencies in the tasks in which they are to be used, they still only model one 'mixture' representation per word, notwithstanding how well it is sculpted.

The following investigations have tackled the challenge of lexical ambiguity head-on with the help of various clustering techniques, which are able to decompose word embeddings into *sense embeddings*. Reisinger and Mooney's seminal work (2010) to vector space word-sense disambiguation introduces the idea of a multi-prototype vector-space model which can be used to learn vectors for different senses of a word. Huang et al. (2012) are the first to use these models to represent the combination of local word and global document context as weighted average vectors, which can then be clustered to form different sense groups and used to learn multi-prototype vectors. The downside to this approach is the need to train the neural embeddings twice, which is computationally expensive, and the embedding methods used are based on an early (less efficient) model by Collobert et al. (2008). A more efficient extension of Huang et al. (2012) is presented in Neelakantan et al. (2015), termed *Multiple Sense Skip-gram*. Reportedly, the model is 27 times faster than approach by Huang et al. (2012) due to the joint performance of word sense discrimination and embedding learning, while also offering a non-parametrical variant of their model, which creates a new cluster "if an observed context is sufficiently different from existing clusters" (Neelakantan et al. 2015: 5).

Whereas the former approaches only rely on unsupervised word sense induction, the approach by Chen et al. (2014) leverages the sense-inventory of WordNet (1995) to learn a fixed set of senses after training an embedding model on a large set of unlabelled data. Finally, we will present the currently most recent contribution by Trask et al. (2016) termed Sense2Vec, which performs supervised clustering for word-sense disambiguated embeddings. In fact, this innovative approach relies on a labelled corpus from which it counts the uses of each unique word (dependent on the labels), generates a random sense

embedding for each use and subsequently predicts a word sense given the surrounding senses. Not only does Sense2Vec provide a fast, accurate and natural way to select a sense-disambiguated embedding, but it is also able to capture more nuanced senses (i.e. sarcasm, figure of speech) than mere part-of-speech confusions.

2.4. Conclusion

In conclusion to this discussion, the needs of and promising roads for embedding research will be presented with respect to the challenge of ambiguity. First of all, experimental studies are required to observe what varieties of features really help for resolving ambiguity in different types of NLP tasks (word vs. phrase-oriented ; syntactically, semantically or lexically inspired …) and analyse how and why increases in performance or accuracy are reported. For example, Levy et al. (2015a) evaluate the effect of supervision for the specific task of lexical inference relations. In so doing, they report the danger of overfitting when supervised methods are used. In general, it has already been observed that the application domain should determine the embedding methods to be used, how they should be adapted to the task and what type of supervision might be helpful. Subsequently, more research is needed in the domain of sense embeddings, particularly to analyse the most efficient way of clustering prototypes or jointly learning senses from context or from external resources. Once the former two have been properly researched, the NLP community should explore how disambiguated embeddings perform in combination with other varieties of supervision and consuming NLP tasks (as suggested by Trask et al. (2016)).

3. The battle for the best context vector representation

Count-based semantic vectors and prediction-based neural embeddings seek to model context in such a way that it is efficient and as much information as possible can be extracted out of a large collection of textual data. Both have been successfully used in NLP and IR tasks that rely on this type of leveraged information. Notwithstanding, recent distributional semantics research has been dominated by research into the newer, more popular embedding type of models.

In what follows, we will sketch an overview of suggested answers to the following questions which lie at the heart of theoretical and experimental research in NLP and IR:

- What are the advantages or disadvantages of a chosen representation approach (either co-occurrence counted or predicted)?
- If so, how do the new neural models improve over the more traditional count-based models?
- For a given specific language task, does one model perform consistently better than the other, and why can we find (large) differences?

A long tradition in NLP and theoretical linguistics has underscored the importance of context in the approximation of word meaning (Harris 1954; Firth 1957; Rumelhart, Hinton and Williams 1986; Miller and Charles 1991). Two approaches can be discerned that each operationalize the vague notion of context in a different way. On the one side, we have the **count-based models**, which mark the occurrences of neighbouring words mapped to vector space, and **predictive models**, which directly predict a word from its neighbours in terms of dense embeddings. Amongst the earliest of methods modelling word co-occurrence statistics we find *Latent Semantic Analysis* (LSA, Deerwester et al. 1990). This popular technique succeeds at modelling context vectors through some form of dimension compression (e.g. singular value decomposition). Another well-known technique in the count-based family is *Latent Dirichlet Allocation* (LDA, Blei et al. 2003), which has mostly gained reputation as a probabilistic topic model. Distributional vectors and the counting, weighting and compressing methods associated with them are highly effective at identifying word similarity in linguistic tasks. However, computational constraints have forced researchers to look for more efficient alternatives.

Within the neural-network community a sequence of papers by Mikolov et al. (2013 a, b) has sparked an overall surge of popularity towards predictive models. More generally, they have devised efficient architectures to train vectors and induce embeddings – distributed, dense low-dimensional representations of words (or larger 'linguistic' units) – from unstructured text data, which in consequence can be directly used as features in other tasks. These recently introduced methods are able to learn high-quality, low-dimensional (typically 50 – 200) vector representations which still manage to automatically capture linguistic patterns present in the data and within the encoded vector space. Not only are word embeddings highly scalable to large datasets, they have also been demonstrated to be efficient as semantic representations due to their compactness and geometry capturing word-interrelationship information.

3.1. Superiority of neural embeddings

Several claims have been made regarding the superiority of the newer type of embedding techniques. For example, Mikolov et al. (2013b) already show in a range of similarity and analogy-reasoning tasks that embeddings perform better than LSA for preserving linear regularities among words. What's more, their neural embeddings were designed for absolute efficiency, presenting improved accuracy at an even lower computational cost. Paradis et al. (2013: 455) argue that "[t]raditional semantic vectors, such as those produced by Latent Semantic Analysis (…), ignore the compositional or deep structure of language". Moreover, the more conventional count methods only operate on surface features and are faced with computational obstacles when scaled to larger datasets and vocabularies.

Although word embeddings present a lot of advantages over more traditional models, they tend to have problems of their own, specifically with regard to training (Weston and Bordes 2014). Another aspect of embedding algorithms illustrated and evaluated in Chen et al. (2013: 1) is that "not all embeddings are created equal". Attributable to their design, including training corpus size and choice of objective function, some embedding techniques perform better on the one task than on the other. By means of a classification task to isolate the effect of context, they have evaluated how well embeddings capture diverse types of information encoded in their geometry. The comparative results expound differences in overall quality and usefulness and points to the idea that the application domain should determine the embedding method.

Current research into word representations is being dominated by the new generation of word embeddings. And rightfully so, according to Baroni et al. (2014) who have conducted a systematic, empirical comparison of the two types of context vectors. Reportedly, the new-and-improved embeddings have been used in a large range of NLP and IR tasks and have re-defined the state-of-the-art or at least improved the near-state-of-the-art. Baroni et al. (2014: 245)'s results emphasize that neural embeddings "are so good that, while the triumphalist overtones still sound excessive, there are very good reasons to switch to the new architecture". However, they do concede that the evaluation focuses on quantitative measures and qualitative studies might shed an interesting light on the errors that the different types of models make.

In the study of Baroni et al. (2014) embeddings have been found superior on NLP tasks concerning semantic relatedness, synonymy, concept categorisation and analogy reasoning. On the other hand, count vector-based models consistently perform better on the selectional restrictions task. Other noteworthy "superior" or impressive results for embeddings have been achieved on Named Entity Recognition (Passos et al. 2014), Unsupervised POS-tagging (Lin et al. 2015), Dependency Parsing (Komatsu et al. 2015), Social Media Sentiment Analysis (Wang and Yang 2015) and Machine Comprehension (Trischler et al. 2016). In the Information Retrieval scene, text embeddings are still being explored in the context of very innovative themes such as cross-lingual retrieval models (Vulic and Moens 2015), induction from clickthrough data (Shen et al. 2014) etc.

3.2. Mere perception of superiority?

All the above considered, we should also point to research papers in embedding literature which sketch a more nuanced view of the "perceived superiority" of embeddings. Hill et al. (2014) re-emphasize that there is not one embedding approach that is best for all tasks. Schnabel et al. (2015) backs up the previous claim with a fine-grained survey of evaluation measures for highlighting specific aspects of embedding performance. They present a novel evaluation framework based on direct comparisons between popular counting and embedding approaches. Again, their results confirm the variability across

embeddings, yet they do stress the previously unacknowledged role of the effect of word frequency on word embedding quality. Even more striking (counter-)evidence is presented by the large experimental study of Levy et al. (2015b). The claims from earlier studies (most notably Baroni et al. 2014) have been put to the test again with different hyperparameter settings ("vanilla scenario", "recommended configuration" and "best configuration adapted to task"). In fact, the controlled experiments reveal no inherent reason for either model to be superior. Moreover, counting methods and predictive methods achieve comparable performance when stripped off of certain implied, pre-tuned parameter settings. Earlier reported performance differences can be attributed to "certain system design choices and hyperparameter optimizations" (Levy et al. 2015b: 1). Moreover, careful hyperparameter tuning has been observed to have a larger impact on the final performance than adding more data (viz. scaling to large datasets). Nevertheless, as has also been observed, larger datasets typically find the optimal hyperparameter configuration. Still, it is deemed practical and beneficial to properly optimize hyperparameter settings for a given task. On a concluding note, Levy et al. (2015b) express the need for controlled-variable, transparent and reproducible experiments in embedding research. In the spirit of the previous studies, Lebret and Collobert (2015) propose a rehabilitation of count vector-based models, which presently have been living in the shadow of the newer context-predicting models. They demonstrate the usefulness (and nice results) of the traditional, simple models on both similarity and analogy tasks, while highlighting the feasibility of count-based context representations. For example, in an earlier paper they induce word embeddings through a spectral technique called *Hellinger PCA* (Lebret and Collobert 2014). The technique adopts a reconstruction approach, which means as much information as possible is captured from the original co-occurrence matrix. It is shown to be competitive with the newer neural network-based context predictive modelling techniques on a sentiment classification task. The above studies exemplify that although neural embeddings have proven to be extremely valuable, counts still matter and can already capture a lot of semantic and syntactic information.

3.3. Conclusion

In conclusion, we will point to some promising roads to explore for the future. Whereas the discussion has focused on the superiority of either type of context vectors, a current trend in embedding research point toward combinations of count and predict methods. As early as Bengio et al. (2003), a mixture model has been hypothesized which is effective because of the difference in kind of mistakes both types of models make. A state-of-the-art paper by Mitra et al. (2016) extends this idea to a *Dual Embedding Space Model* for document ranking. In IR and subsequent relevance ranking it is important to differentiate between a document's "aboutness" or "mentioning" of a term. Count-based models tend to overlook this essential dichotomy. However, the paper proposes the combination of term frequency and occurrence of other related terms as evidence of "aboutness". In fact, LSA's train on the word-document matrix, which is better for capturing global context, and

word2vec embeddings train on word co-occurrence data within a given window, thus better fit for enclosing local contextual information. Relatedly, Christopher Moody has worked out a hybrid algorithm, *lda2vec*, which combines the best from LDA ("interpretable" topic modelling) and word2vec (originally language modelling for powerful and flexible representations). The framework aims to build topic models exploiting the best properties of both algorithms. It has been successfully employed to mine client-item descriptions (Moody 2016) and is being researched to scale to sentences in an extension, *lda2stm*.

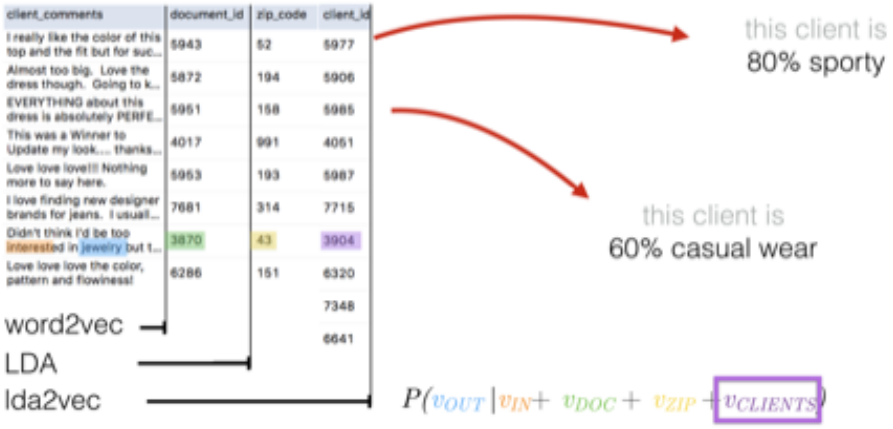

Figure 1: An illustration of lda2vec in the context of client item descriptions.
It is able to yield topics over documents, regions and even clients.

Let it be clear from the previously illustrated success stories that further research in combining different count and predict techniques holds promises for advancements in NLP and IR.

Formulating a straightforward answer to the questions presented at the start of the discussion has proven to be more difficult than expected. There is a lot of disagreement in word representation research concerning the superiority of the neural-network modelling techniques. In my humble opinion, I believe the second, more nuanced group to be more correct in the approach of these questions. Certainly Levy et al. (2015b) make a viable point concerning the need for transparency and controlled-variable experiments in embedding research. However, neither do I believe that all reported performance differences simply boil down to hyperparameter optimization or a lack thereof. Both types of context vector approaches have their advantages and disadvantages and should therefore be exploited as such. Nevertheless, the recent impact of neural embeddings on the fields of NLP and IR should not be downgraded. They will form an important building block for future applications, yet a lot of directions still deserve thorough investigation so we can benefit optimally from these remarkable techniques. For example, Levy et al. (2015a) have shown that context vectors are "inherently handicapped in capturing relational information, requiring supervised methods to harness complementary information from more sophisticated features". In truth, it should be recognised that some supervision adapted to the specific task will be needed to effectively see an application relying on context vectors through. Moreover, more complex learning strategies (joint/sequential) should be

researched to scale up to larger linguistic/textual units. Task-specific embedding approaches exploring either or both neural embeddings and/or traditional counting models will certainly boost theoretical and experimental research in NLP and IR.

4. Bibliography

"fluke", *OED Online*. Oxford University Press, March 2016. Web. 12 April 2016.

"Linguistic problems and complexities", *School of Electrical Engineering and Computer Science Uottawa*, March 2016. http://www.csi.uottawa.ca/tanka/files/complexities.html

Baroni, M., Dinu, G. and Kruszewski, G. 2014. Don't count, predict! a systematic comparison of context counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), pages 238{247, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

Bengio, Yoshua, Ducharme, Réjean, Vincent, Pascal, and Janvin, Christian. 2003. A neural probabilistic language model. J. *Mach. Learn. Res.*, 3:1137–1155, March 2003. ISSN 1532-4435.

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

Chen, Tao, Xu, Ruifeng, He, Yulan, and Wang, Xuan. 2015 Improving distributed representation of word sense via wordnet gloss composition and context clustering". In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (Volume 2: Short Papers), pp. 15–20, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-2003.

Chen, Xinxiong, Liu, Zhiyuan, and Sun, Maosong. 2014. A unified model for word sense representation and disambiguation. In Proceedings of the 2014 *Conference on Empirical Methods in Natural Language Processing* (EMNLP), pp. 1025–1035, 2014.

Chen, Y., Perozzi, B., Al-Rfou, R.' and Skiena, S., 2013. The expressive power of word embeddings. *CoRR*, abs/1301.3226.

Collobert, R. and Weston, J., 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In International Conference on Machine Learning, ICML.

Cross, J., Xiang, B. and Zhou, B., 2015. Good, Better, Best: Choosing Word Embedding Context. *arXiv preprint arXiv:1511.06312*.

Deeplearning4j Development Team. 2015 Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. URL: http://deeplearning4j.org

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. & Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41 (6), 391- 407.

Fellbaum, C., 1999. The organization of verbs and verb concepts in a semantic net. In *Predicative forms in natural language and in lexical knowledge bases* (pp. 93-109). Springer Netherlands.

Firth, John R. 1957. A synopsis of linguistic theory 1930–1955. In *Studies in linguistic analysis*, 1–32. Oxford: Blackwell.

Ganguly, D., Roy, D., Mitra, M., and G. J. Jones. 2015. Word embedding based generalized language model for information retrieval. In *Proc. SIGIR*, pages 795–798. ACM.

Ganitkevitch, J., Van Durme, B. and Callison-Burch, C., 2013. PPDB: The Paraphrase Database. In *HLT-NAACL* (pp. 758-764).

Gormley, Matthew R., Yu, Mo and Dredze, Marc. 2015. Improved Relation Extraction with Feature-Rich Compositional Embedding Models. *Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

Harris, Zellig. 1954. Distributional structure. Word 10(23). 146–162.

Hill, F., Cho, K., Jean, S., Devin, C. and Bengio, Y., 2014. Not all neural embeddings are born equal. arXiv preprint arXiv:1410.0718, 2014.

Hinton, G.E., McClelland, J.L., Rumelhart, D.E. 1986. Distributed representations. Parallel distributed processing: Explorations in the microstructure of cognition, 1(3):77–109.

Huang, Eric H., Socher, Richard, Manning, Christopher D., and Ng, Andrew Y. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers* - Volume 1, ACL '12, pp. 873–882, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2390524.2390645.

Jurafsky, Daniel, and James H. Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. 2nd edition. Prentice-Hall.

Komatsu, H., Tian, R., Okazaki, N. and Inui, K., 2015. Reducing Lexical Features in Parsing by Word Embeddings.

Lebret, R. Collobert, R., 2014. Word Embeddings through Hellinger PCA. *In EACL.*

Lebret, R. Collobert, R., 2015. Rehabilitation of Count-based Models for Word Vector Representations. *In CICLing.*

Lederer, Richard 1998. *The Word Circus*. Merriam-Webster. p. 23. ISBN 0877793549.

Levy, O. and Goldberg, Y., 2014. Dependency-Based Word Embeddings. In *ACL (2)* (pp. 302-308).

Levy, O., Goldberg, Y. and Dagan, I., 2015b. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, *3*, pp.211-225.

Levy, O., Remus, S., Biemann, C., Dagan, I. and Ramat-Gan, I., 2015a. Do supervised distributional methods really learn lexical inference relations. *Proceedings of NAACL, Denver, CO*.

Lin, Chu-Cheng, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised POS induction with word embeddings. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. Denver, CO, pages 1311–1316.

Ling, Wang, Dyer, Chris, Black, Alan W, and Trancoso, Isabel. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1299–1304, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1142.

Mikolov T., Sutskever I., Chen K., Corrado G., and Dean J., 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th Annual Conference on Advances in Neural Information Processing Systems* (NIPS), pp. 3111-3119.

Mikolov T., Sutskever I., Chen K., Corrado G., and Dean J., 2013a. "Efficient estimation of word representations in vector space," CoRR, vol. abs/1301.3781.

Miller, G.A. and Charles, W.G., 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, *6*(1), pp.1-28.

Moody, Christopher 2016, image of lda2vec in progress, https://github.com/cemoody/lda2vec/blob/master/images/img04_lda2vec_topics02.png

Moody, Christopher. 2016. "lda2vec: introducing a new hybrid algorithm". http://www.slideshare.net/ChristopherMoody3/word2vec-lda-and-introducing-a-new-hybrid-algorithm-lda2vec

Moody, Christopher. 2016. lda2vec: freely distributed. URL: https://github.com/cemoody/lda2vec

Nalisnick, E. and Ravi, S., 2015. Infinite Dimensional Word Embeddings.*arXiv preprint arXiv:1511.05392*.

Nalisnick, E., Mitra, B., Craswell, N. and Caruana, R., 2016 (April). Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference Companion on World Wide Web* (pp. 83-84). International World Wide Web Conferences Steering Committee.

Neelakantan, Arvind, Shankar, Jeevan, Passos, Alexandre, and McCallum, Andrew. 2015. Efficient nonparametric estimation of multiple embeddings per word in vector space. *CoRR*, abs/1504.06654, 2015. URL http://arxiv.org/abs/1504.06654.

Paradis, R.D., Guo, J.K., Moulton, J., Cameron, D. and Kanerva, P., 2013. Finding semantic equivalence of text using random index vectors. *Procedia* Computer Science, 20, pp.454-459.

Passos, A., Kumar, V. and McCallum, A., 2014. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*.

Pennington, J., Socher, R., and Manning, C. 2014, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (EMNLP), pp. 1532-1543.

Princeton University "About WordNet." WordNet. Princeton University. 2010. http://wordnet.princeton.edu

Reisinger, Joseph and Mooney, Raymond J., 2010. Multi-prototype vector-space models of word meaning. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10, pp. 109–117, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL http://dl.acm.org/citation.cfm?id=1857999.1858012.

Schnabel, T., I. Labutov, D. Mimno, and T. Joachims. Evaluation methods for unsupervised word embeddings. In Proc. EMNLP, 2015.

Shen, Y., He, X., Gao, J., Deng, L., and Mesnil G., 2014. Learning semantic representations using convolutional neural networks for web search. In Proc. WWW, pages 373–374.

Trask, A., Michalak, P., Liu, J., 2016. sense2vec - A Fast and Accurate Method for Word Sense Disambiguation In Neural Word Embeddings. *arXiv:1511.06388* [cs].

Trischler, A., Ye, Z., Yuan, X., He, J., Bachman, P., Suleman, K., 2016. A Parallel-Hierarchical Model for Machine Comprehension on Sparse Data. *ArXiv e-prints* 1603, arXiv:1603.08884.

Vulic, I. and Moens, M.-F. 2015. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proc. SIGIR*, pages 363–372. ACM.

Wang, W.Y. and Yang, D., 2015(Sept). That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), Lisbon, Portugal*.

Weston, J. and Bordes, A. 2014. "Embedding Methods for NLP: Part 1: Unsupervised and Supervised Embeddings" http://emnlp2014.org/tutorials/8_notes.pdf

Yu, Mo and Dredze, Mark. 2015. Learning Composition Models for Phrase Embeddings. *Transactions of the Association for Computational Linguistics*, 2015.

Yu, Mo, Gormley, Matthew R. and Dredze, Marc. 2014. Factor-based Compositional Embedding Models. *NIPS Workshop on Learning Semantics*.