International Workshop on Embeddings and Semantics

SEPLN ‘15

Parth Gupta, Rafael E. Banchs and Paolo Rosso

9-15-2015

Program Committee

Eneko Agirre, University of the Basque Country
Toni Badia, Universitat Pompeu Fabra
Marco Baroni, University of Trento
Núria Bel, Universitat Pompeu Fabra
Davide Buscaldi, Université Paris XIII
Monojit Choudhury, Microsoft Research
Montse Cuadros, Vicomtech
Marc Franco-Salvador, PRHLT, Universitat Politècnica de València
German Kruszewski, University of Trento
Antonia Marti, Universitat de Barcelona
Tomas Mikolov, Facebook
Alessandro Moschitti, Qatar Computing Research Institute
Roberto Navigli, Sapienza University of Rome
Lluís Padró, Universitat Politècnica de Catalunya
Martha Palmer, University of Colorado Boulder
Nghia Pham, University of Trento
German Rigau, University of the Basque Country
Horacio Rodriguez, Universitat Politècnica de Catalunya
Marta Ruiz, Instituto Politécnico Nacional
Sakriani Sakti, NAIST
Dan Tufis, RACAI
Michael Zock, Sapienza University of Rome

Preface

Parth Gupta and Paolo Rosso
PRHLT Research Center
Universitat Politècnica de València, Spain
[pgupta,prosso]@dsic.upv.es

Rafael E. Banchs
HLT, Institute for Infocomm Research
Singapore
rembanchs@i2r.a-star.edu.sg

Abstract: The main objective of the International Workshop on Embeddings and Semantics (IWES) is to bring together researchers interested in the use of continuous space embeddings for modelling language, semantics and meaning. Recent advances on neural networks and related machine learning applications have proven the use of continuous spaces to be useful for natural language applications in both the monolingual and cross-language settings. The topics of the workshop include, but are not restricted to:

- latent semantics for natural language processing
- properties and applications of continuous space representations of language
- use of embeddings for multimodal semantics
- deep learning for language modelling
- cross-language information retrieval and information extraction

The workshop had contained a keynote speech with title “From conceptual to referential properties with distributional semantics” by Gemma Boleda and 7 paper presentations. The selected papers cover a wide range of technical contributions to embeddings and semantics. Some of them are showing the benefits of embeddings for applications such as definition extraction, machine translation, medical vocabulary expansion and topic models for short-text. Some others present modelling aspects of the embeddings like generative modelling, multimodal embeddings and word-order sensitive vector space models. The workshop had also contained panel discussion on the current state and future directions on embeddings and semantics.

Keynote: From conceptual to referential properties with distributional semantics

Gemma Boleda
University of Trento
Italy
gemma.boleda@upf.edu

Abstract: Distributional semantics (including representations known as "word embeddings") is a very successful, radically empirical, scalable, and flexible approach to meaning. Its representations account for generic properties that are akin to conceptual knowledge: Cats are similar to dogs, semantically related to veterinaries, and not very plausible agents of flying. However, when it comes to referring to a particular cat with specific properties, distributional semantics doesn't fare very well --and yet, reference is crucial to language, since we use words to talk about things in the world. I will review some referential phenomena that a more comprehensive model of meaning should handle, and discuss some theoretical and empirical work towards modelling reference with distributional semantics. In particular, I will show that word embeddings encode some referential properties of real-world entities such as countries and cities, and discuss the potential and limitations of learning a mapping between conceptual (distributional) and referential (database) representations.

Deep Autoencoder Topic Model for Short Texts

Modelos de T'opicos para textos cortos mediante auto-codificadores de m'ultiples capas

Girish Kumar
NUS High School of Math and Science
Singapore 129957
girishvilla@gmail.com

Luis F. D'Haro
Institute for Infocomm Research
Singapore 138632
luisdhe@i2r.a-star.edu.sg

Resumen: En este trabajo presentamos un método para modelado de tópicos mediante el uso de auto-codificadores de múltiples capas (DATM por sus siglas en inglés). El objetivo principal de estos modelos es la extracción de distribuciones de tópicos en textos cortos. En un análisis comparativo, el método propuesto proporciona mejores resultados que otros métodos convencionales (LSA y LDA).

Palabras clave: Modelos de tópicos, Máquinas de Boltzmann Restringidas

Abstract: We present the Deep Autoencoder Topic Model (DATM) for the purpose of discovering topics from short texts. The DATM is trained in two steps: i) greedy layer-wise pre-training as Sparse & Selective Restricted Boltzmann Machines (RBMs) and ii) parameter fine-tuning with back-propagation. When benchmarked with the topic coherence metric, the DATM outperformed Latent Semantic Analysis and Latent Dirichlet Allocation.

Keywords: Topic models, restricted boltzmann machines, autoencoders, deep learning

1 Introduction

Topic models discover hidden topical structure in large sets of training documents by assuming that hidden topics (latent variables) generate the training documents (observed variables) through a generative process. Topics are usually described by a set of related words over a fixed vocabulary. Prominent topic modelling methods include Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). An introduction to LSA and LDA is given in the Appendix.

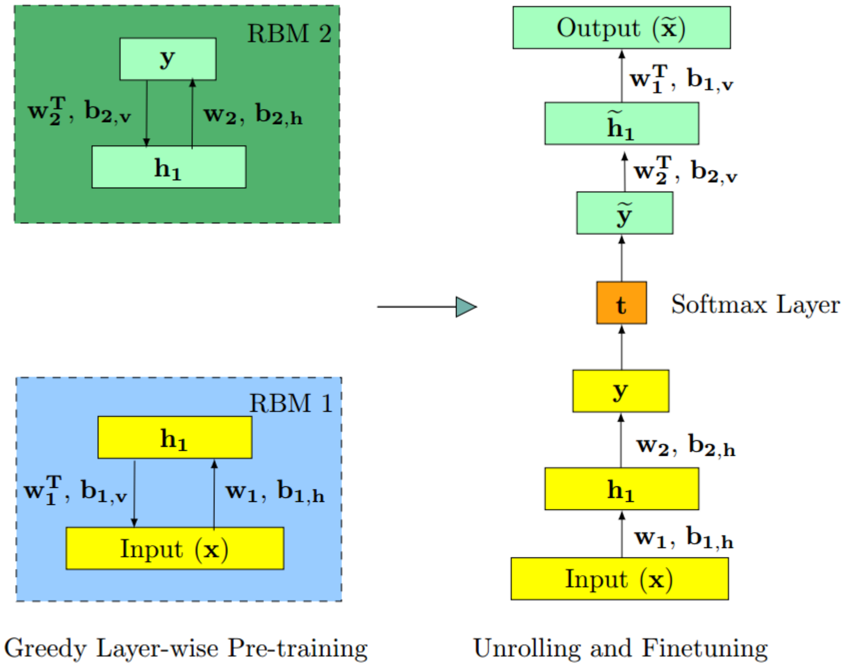
However, LDA and LSA do not perform well on short documents (in our case, sentences) as they do not model word-word co-occurrences well. As such, we propose a novel Deep Auto-encoder Topic model (DATM) that models both word-word and word-document co-occurrences.

2 Methodology

Figure 1 provides an overview of DATM training. To capture word-word and word-document co-occurrences, we chose the training input to be document vectors, $\mathbf{x} \in [0,1]^n$ where n is the vocabulary size. The DATM is then trained on input documents in two steps: i) greedy layer-wise pre-training as generative Restricted Boltzmann Machines (RBMs) and ii) parameter fine-tuning with back-propagation to learn the identity approximation of the input data for dimensionality reduction.

First, each RBM is trained greedily (left of Figure 1) using contrastive divergence (CD) learning [Hinton *et al.*2006b]. A short introduction to RBMs and contrastive divergence learning is provided in the Appendix. The RBMs consist of stochastic, binary units and are trained one by one starting from the bottom-most RBM which directly takes the input data. The upper RBMs take the output of the trained RBM below. The goal of CD learning is similar to that of LDA: tuning the RBM's weights $\mathbf{w}$ and biases $\mathbf{b}$ to find the set of latent variables, $\mathbf{h_1}, \mathbf{y}$, that maximise the probability of observing the documents. After RBM pre-training, the DATM is unrolled (right of Figure 1) to reconstruct the input data vectors for dimensionality reduction and to map topics to their constituent words. A softmax bottleneck layer is added to normalize the hidden output, $\mathbf{y}$, of the RBMs to a probability distribution. Note that each unit in the soft-max layer corresponds to each latent topic to be discovered. The stochastic binary activities of the other feature layers are replaced by the real-valued probabilities. To fine-tune the network, we back-propagate the gradient of the mean cross-entropy error (E) between $\widetilde{\mathbf{x}}$ and $\mathbf{x}$. d is the number of documents.

$$E(\widetilde{\mathbf{x}}, \mathbf{x}) = -\frac{1}{d}\sum_{k=1}^{d}[\mathbf{x}_k \log \widetilde{\mathbf{x}}_k + \\ (1-\mathbf{x}_k)(1-\log \widetilde{\mathbf{x}}_k)] \tag{1}$$


RBM 2
$\mathbf{y}$
$\mathbf{w_2^T}$, $\mathbf{b_{2,v}}$
$\mathbf{w_2}$, $\mathbf{b_{2,h}}$
$\mathbf{h_1}$
RBM 1
$\mathbf{h_1}$
$\mathbf{w_1^T}$, $\mathbf{b_{1,v}}$
$\mathbf{w_1}$, $\mathbf{b_{1,h}}$
Input ($\mathbf{x}$)
Output ($\tilde{\mathbf{x}}$)
$\mathbf{w_1^T}$, $\mathbf{b_{1,v}}$
$\tilde{\mathbf{h}}_\mathbf{1}$
$\mathbf{w_2^T}$, $\mathbf{b_{2,v}}$
$\tilde{\mathbf{y}}$
$\mathbf{t}$
Softmax Layer
$\mathbf{y}$
$\mathbf{w_2}$, $\mathbf{b_{2,h}}$
$\mathbf{h_1}$
$\mathbf{w_1}$, $\mathbf{b_{1,h}}$
Input ($\mathbf{x}$)

Greedy Layer-wise Pre-training

Unrolling and Finetuning

Figure 1: DATM Training Overview

Upon DATM training, we obtained the topic distribution, $\mathbf{t}$, of a document by first vectorizing it and then computing the soft-max layer activations (right of Figure 1, bottom part). For obtaining the words that describe each topic, we found words in the vocabulary that are associated with the activations of each softmax layer unit (right of Figure 1, upper part). To find the words that describe the k^{th} topic, we strongly activate the corresponding k^{th} soft-max unit by letting $\mathbf{t}[i] = \begin{cases} 0 & : i \neq k \\ 1 & : i = k \end{cases}$. We then compute the output activations with $\mathbf{t}$ and obtain the words that correspond to the output units with the highest activations.

2.1 Sparse and Selective RBMs

During testing, we found that all the topics decoded from the auto-encoder consisted of the exact same words. Interestingly, we found that these words were also the most frequently occurring terms as training was stuck in an undesirable local minimum of the cost function. Closer inspection found all of the hidden layer neurons being activated regardless of the input. We hypothesized that this was due to the sparsity of the input document vectors due to the short length of sentences. To solve this issue, inspired by [Lee *et al.*2008], we modified the RBM cost function to include sparsity and selectivity penalty terms. Sparsity ensures that each document belongs to at most a few topics. Selectivity ensure that each topic encodes for only a subset of the training documents. Hence, given d training examples $\{\mathbf{v}^{(1)}, ..., \mathbf{v}^{(d)}\}$, training a sparse and selective RBM, with n hidden units $\mathbf{h}$, is presented in the form of an optimization problem, as defined in Equation 2.1.

$$\text{minimize}_{w_i,b_{i,h},b_{i,v}}\{\underbrace{-\frac{1}{d}\sum_{l=1}^{d}\sum_{\mathbf{h}}\log(p(\mathbf{v}^{(l)}))}_{\text{RBM Log Likelihood}} + \underbrace{\lambda\cdot\frac{1}{n}\sum_{j=1}^{n}|\rho-\frac{1}{d}\sum_{l=1}^{d}\mathbb{E}[h_j^{(l)}|\mathbf{v}^{(l)}]|^2}_{\text{Selectivity Penalty}} + \underbrace{\mu\cdot\frac{1}{d}\sum_{l=1}^{d}|\tau-\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[h_j^{(l)}|\mathbf{v}^{(l)}]|^2\}}_{\text{Sparsity Penalty}} \quad (2)$$

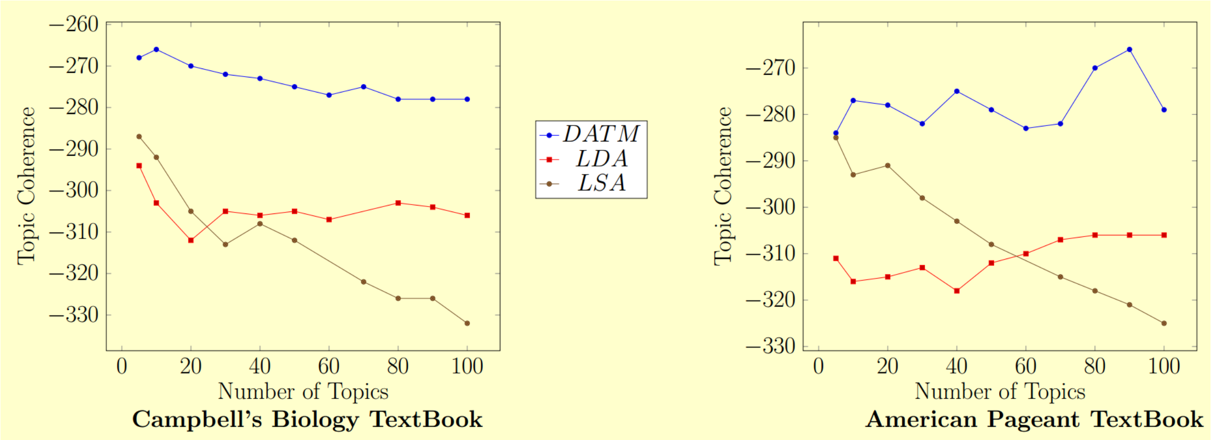


Figure 2: Average Topic Coherence for the various datasets and topic models

where $\mathbb{E}[h_j^{(l)}|\mathbf{v}^{(l)}]$ is the expected activation of hidden unit h_j given input $\mathbf{v}^{(l)}$; ρ, τ are the selectivity & sparsity targets and λ, μ are the penalty-term weights. Since computing the log-likelihood gradient is intractable, we deal with minimising the log-likelihood term and the penalty terms separately. Contrastive divergence was used to estimate the log-likelihood gradient to update the weights and the biases. Gradient descent is used for the penalty terms to update only the biases as they directly control the degree to which the hidden neurons are activated [Lee *et al.*2008].

3 *Experimentation & Results*

For benchmarking the proposed DATM with LDA and LSA, we used 2 corpora for which statistics are in Table 1. Campbell's Biology and The American Pageant are high school textbooks for biology and US history respectively. Here, each sentence is a document. The textbook datasets will allow us to to benchmark the performance of the DATM on short and informative sentences which is relevant to our work. We preprocessed all the datasets by removing stopwords and stemming. Also, we only consider the 2000 most frequent words in each dataset.

	Campbell Biology	**American Pageant**
No. of Documents	35621	22797
Words per Document	20	19

Table 1: Corpora Used for DATM Benchmarking & Evaluation

We use Average Topic Coherence (ATC) for performance benchmarking [Mimno *et al.*2011]. ATC computes a sum of pair-wise scores on the top n words, w, that describe a topic.

$$\text{ATC} = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=2}^{n}\sum_{j=1}^{i-1}\log\frac{D(w_i, w_j)+1}{D(w_i)} \tag{3}$$

where T is the number of topics while $D(w_i)$ and $D(w_i, w_j)$ are the counts of training documents containing the word w_i and both the words w_i and w_j respectively. A better topic model will result in a less negative ATC. ATC was chosen as it was found to be strongly correlated to human judgement of topics [Mimno *et al.*2011]. Furthermore, ATC quantifies the extent to which the topic model captures word-word co-occurrences. Figure 2 shows the ATC scores for the DATM, LDA and LSA for the 2 datasets. Each topic model was used to discover various number of topics from each dataset, ranging from 5 to 100.

The LDA model was trained using the implementation in the Matlab Topic Modelling Toolbox [Griffiths and Steyvers2004]. 300 iterations were used for training on all the corpora. The *gensim* package was used for LSA [Řehůřek and Sojka2010]. We implemented our proposed DATM with Theano [Bastien *et al.*2012]. A hidden layer size of $[500]$ and sparsity $=$ selectivity $=$ 0.03 were used. 50 iterations were used for training on all the corpora. Note that we chose parameters based on those proposed by Hinton et. al [Hinton2010b]. For ATC calculations, the top 20 words for each topic were used.

Evidently, our proposed DATM outperforms LDA and LSA on the ATC metric. However, better performance on ATC does not conclusively prove DATM's superiority. More tests with multiple performance metrics are required to do so. Nonetheless, the results do manifest that DATM is better in modelling word-word co-occurrences.

4 Conclusion

In this paper, with the aim of topic modelling short and informative texts, we have proposed the Deep Autoencoder Topic Model (DATM). Training the DATM consists of two steps: 1) Greedy Layer-wise Pre-training & 2) Unrolling and fine-tuning via backpropagation. Furthermore, to deal with the issue of the sparsity, we added sparsity and selectivity penalties to the RBM cost function. The DATM was finally benchmarked with the topic coherence metric with textbook datasets, where it outperformed the widely-used LDA (Latent Dirichlet Allocation) and LSA (Latent Semantic Analysis).

Acknowledgement

The authors would like to thank Dr. Rafael E. Banchs from the Institute for Infocomm Research for his advice and support during the realization of this work.

References

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

Y. Bengio. Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, November 2009.

Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

G. E. Hinton, S. Osindero, and Y-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 2006.

Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

G. E. Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 2010.

Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926, 2010.

J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Biophysics*, 1982.

Honglak Lee, Chaitanya Ekanadham, and Andrew Y Ng. Sparse deep belief net model for visual area v2. In *Advances in neural information processing systems*, pages 873–880, 2008.

David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics, 2011.

Radim Řehůřek and Petr Sojka. 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 2010. ELRA.

R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, 2009.

A Appendix

A.1 LSA

Latent Semantic Analysis (LSA) is one of the most widely-used methods for learning latent topics from text and is often used for dimensionality reduction. Given a document-term matrix, $\mathbf{M} \in \mathbb{R}^{V \times N}$, where V is the number of words in the vocabulary and N is the number of input training documents, LSA factorizes $\mathbf{M}$ using Singular Value Decomposition to find a low-rank approximation given as follows. Rank lowering results in the combination of some dimensions which results in dependence on more than one term.

$$\mathbf{M} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

$\mathbf{U}$ and $\mathbf{V}$ represent word and document embeddings on the latent topic space.

A.2 LDA

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a set of documents. Documents are represented as random mixtures over hidden topics, where each topic is characterized by a distribution over words. The following generative process is assumed for each document in a corpus.

1. Choose number of words, N ~Poisson(μ)
2. Choose topic mixture/distribution θ ~Dirichlet(α)
3. Choose topics z_k ~Dirichlet(θ)
4. Choose words w ~Multinomial(ϕ_k)

Expectiation-Maximization or Gibbs Sampling can then be utilized for inferring topics from the assumed generative process.

A.3 Restricted Boltzmann Machines (RBMs)

Restricted Boltzmann Machine (RBM), a bipartite graph variant of the boltzmann machine, is an energy-based probability model to infer hidden variables [Bengio2009]. The bipartite nature of the RBM means that it does not allow connections among units in each layer [Salakhutdinov and Hinton2009], which makes it efficient in learning [Bengio2009]. As a special form of the general second-order polynomial, the energy function of the RBM, formed by the joint configurations of both visible and hidden units $(\mathbf{v}, \mathbf{h})$, is given by [Hopfield1982]:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_i c_i v_i - \sum_j b_j h_j - \sum_{i,j} w_{ij} v_i h_j \tag{4}$$

where **v** is the visible inputs, **h** consists of the hidden nodes or latent variables, i represents the number of dimensions for each input, and j represents the number of hidden nodes. RBMs are trained as probabilistic models by maximizing a log of the following likelihood of the visible vector [Hinton2010a].

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{5}$$

where the partition function, Z, is given as follows

$$Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{6}$$

Contrastive divergence (CD) is used to efficiently approximate the log-likelihood gradient of RBMs [Hinton *et al.*2006a]. The RBM learns in an unsupervised fashion with a stochastic element being introduced in the random sampling process. The CD algorithm updates the weights as in the following:

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \epsilon(\mathbf{h}_t \mathbf{v}_t - \mathbf{h}_{t+1} \mathbf{v}_{t+1}) \tag{7}$$

where the subscript t represents the number of iterations, **v** is the visible inputs, **h** is the hidden vector, and ϵ is the learning rate.

Definition Extraction Using Sense-Based Embeddings

Luis Espinosa-Anke Horacio Saggion
TALN DTIC
Universitat Pompeu Fabra
Carrer Tànger 122-134
08018 Barcelona, Spain
{luis.espinosa,horacio.saggion}@upf.edu

Claudio Delli Bovi
Dipartimento di Informatica
Sapienza Università di Roma
Viale Regina Elena 295
00161 Roma, Italy
dellibovi@di.uniroma1.it

Abstract: Definition Extraction is the task to identify snippets of free text in which a term is defined. While lexicographic studies have proposed different definition typologies and categories, most NLP tasks aimed at revealing word or concept meanings have traditionally dealt with lexicographic (encyclopedic) definitions, for example, as a prior step to ontology learning or automatic glossary construction. In this paper we describe and evaluate a system for Definition Extraction trained with features derived from two sources: Entity Linking as provided by Babelfy, and semantic similarity scores derived from sense-based embeddings. We show that these features have a positive impact in this task, and report state-of-the-art results over a manually validated benchmarking dataset.

Keywords: Embeddings, Entity Linking, Definition Extraction, Information Extraction

1 Introduction

Definitions are fundamental sources for retrieving the meaning of terms (Navigli and Velardi, 2010). However, looking them up manually in naturally occurring text is unfeasible. For this reason, automatic extraction of definitional text snippets is on demand, especially for tasks like Ontology Learning (Velardi, Faralli, and Navigli, 2013; Snow, Jurafsky, and Ng, 2004; Navigli and Velardi, 2006), Question Answering (Saggion and Gaizauskas, 2004; Cui, Kan, and Chua, 2005), Glossary Creation (Muresan and Klavans, 2002; Park, Byrd, and Boguraev, 2002), or support for eLearning environments (Westerhout and Monachesi, 2007).

The task to automatically identify definitions in free text is Definition Extraction (DE). As in many extraction tasks in NLP, a great deal of previous work has relied on linguistic patterns. For instance, by directly identifying verbal cue phrases (Rebeyrolle and Tanguy, 2000; Saggion and Gaizauskas, 2004; Sarmento et al., 2006; Storrer and Wellinghoff, 2006). Moreover, machine learning approaches have incorporated linguistic patterns as information for training classifiers. For instance, (Navigli and Velardi, 2010) propose a generalization of word lattices for the tasks of DE and Hypernym Extraction. In addition, (Boella et al., 2014) exploit syntactic dependencies to create word representations, which are used as features for training an SVM classifier. Moreover, (Jin et al., 2013) use hand-crafted shallow parsing patterns in a CRF-based sequential labeller for DE in scientific papers. Finally, (Espinosa-Anke and Saggion, 2014) take advantage of syntactic dependencies in the form of a bag-of-subtrees approach together with metrics exploiting the dependency tree such as a word's degree or the part-of-speech of its children.

Although the systems reported above achieve competitive results, in none of them semantic information is used, opening therefore clear avenues for improvement. We hypothesize that external knowledge can contribute dramatically to the DE task, and can be also useful for potential cross-domain or multilingual experiments. In this paper, rather than introducing knowledge from structured resources, we leverage SensEmbed (Iacobacci, Pilehvar, and Navigli, 2015), a recent work that applies state-of-the-art representation techniques for modelling individual word senses. Our choice stems from the intuition that sense-based representations can reveal properties of *semantic compactness*, which may be indicators of definitional or gloss-like text snippets.

In the next section we proceed to describe our approach to DE.

2 DE Using SensEmbeddings

2.1 Data

We perform our experiments on the WCL dataset (Navigli, Velardi, and Ruiz-Martínez, 2010), a subset of Wikipedia containing 1717 definitions (coming from the first sentence of randomly sampled Wikipedia articles), and 2847 of what the authors called "syntactically plausible false definitions", i.e. sentences with a syntactic structure similar to that of a definition, and where the defined term appears explicitly, but are not definitions.

2.2 Entity Linking

The first step of our approach consists in running Babelfy (Moro, Raganato, and Navigli, 2014), a state-of-the-art WSD and Entity Linking tool, over the WCL dataset. In this way, we obtain disambiguations for content text snippets, which are used to build a semantically rich representation of each sentence. Consider the following definition and its concepts, represented with their corresponding BabelNet (Navigli and Ponzetto, 2012) synset id:

The⟨O⟩ Abwehr⟨01158579n⟩ was⟨O⟩ a⟨O⟩ German⟨00103560a⟩ intelligence⟨00047026n⟩ organization⟨00047026n⟩ from⟨O⟩ 1921⟨O⟩ to⟨O⟩ 1944⟨O⟩.

This disambiguation procedure yields two important pieces of information. On one hand, the set of concepts, represented as BabelNet synsets, e.g. the synset with id bn:01158579n for the concept Abwehr$_{bn}$[1]. On the other hand, we also obtain a set of non-disambiguated snippets (either single word or multiword terms), which can be also used as indicators for spotting a definitional text fragment in a corpus (from the above example: {*the*, *was a*, *from 1921 to 1944*}).

2.3 Sense-Based Embeddings

SensEmbed works in two main steps: First, a large text corpus is disambiguated with Babelfy. Then, *word2vec* (Mikolov, Yih, and Zweig, 2013; Mikolov et al., 2013) is applied to the disambiguated corpus, yielding a vectorial latent representation of word senses. This enables a disambiguated vector representation of concepts. For instance, for the term "New York" (BabelNet id bn:00041611n), there are vectors for lexicalizations such as "NY", "New York", "Big Apple" or even "Fun City".

We use SensEmbed for computing the semantic similarity among concepts in each sentence of the WCL corpus. These similarities are afterwards used for computing features that will serve as input for a sentence-based classifier. We denote in the rest of this paper the semantic similarity between two concepts x and y as $\textsc{sim}(x, y)$, which is simply the cosine similarity of the closest vectors associated to their corresponding lexicalizations. Formally, let L be the set of lexicalizations included in SensEmbed and Γ the set of associated vectors to each lexicalization. We compute $\textsc{sim}$ as follows: (1) Retrieve all the available lexicalizations in L of both x and y, namely $L(x) = \{s_x^1, ..., s_x^m\}$ and $L(y) = \{s_y^1, ..., s_y^z\}$. (2) Next, retrieve from Γ the corresponding sets of vectors $V(x) = \{v_x^1, ..., v_x^m\}$ and $V(y) = \{v_y^1, ..., v_y^z\}$. (3) Finally, we compare each possible pair of senses and select the one maximizing the cosine similarity $\textsc{cos}$ between the corresponding vectors, i.e.

$$\textsc{cos}\,(x, y) = \max_{v_x \in V(x), v_y \in V(y)} \frac{v_x \cdot v_y}{||v_x||||v_y||}$$

For example, given the definition of the term *bat*, "A bat is a mammal in the order Chiroptera", we obtain a set D of three concepts: bat$_{bn}$, mammal$_{bn}$ and Chiroptera$_{bn}$. For each pair of concepts $c_1, c_2 \in D$, we compute $\textsc{sim}(c_1, c_2)$, and perform this operation over all pairs in D.

Table 1 shows the $\textsc{sim}$ representation of this definition (d) and one non-definitional sentence (n) also referring to *bat*: "This role explains environmental concerns when a bat is introduced in a new setting". Note the higher $\textsc{sim}$ scores for concept pairs in the definitional sentence (in bold). Also, note that since the non-definition is less *semantically compact*, our procedure assigned to the term *bat* vectors corresponding to the programming language *batch*, or to *batch* files.

[1] For clarity, we use the subscript *bn* to refer to the concept's BabelNet id, rather than using the actual numeric id.

Vector	Vector'	SIM
bat$_d$	**mammal**$_d$	0.59
bat$_d$	**chiroptera**$_d$	0.29
mammal$_d$	**chiroptera**$_d$	0.31
role$_n$	environmental_concern$_n$	0.21
purpose$_n$	batch_language$_n$	0.15
environmental_concern$_n$	role$_n$	0.21
conservation_group$_n$	batch_file$_n$	0.12
batch_language$_n$	purpose$_n$	0.15
batch_file$_n$	conservation_group$_n$	0.12

Table 1: Representation of a definition and a non-definition in terms of the similarities of its concepts.

In the remainder of the paper, the whole set of similarity scores over a given sentence, obtained with this strategy, is denoted as Δ.

2.4 Features

We design three types of features: (1) Bag-of-Concepts; (2) Bag-of-non-disambiguated text snippets; and (3) Similarity metrics over Δ. These features are then used to train different classification algorithms, whose performance is evaluated in 10-fold cross validation.

Bag-of-Concepts

We extract the 100 most frequent BabelNet synsets in the training data, and generate a feature vector for each one. Each feature has a binary value, either *True* or *False*, referring to whether such synset was found in the sentence to be classified. In most folds, the most frequent synsets refer to ancient languages such as Greek or Latin, or to scientific disciplines such as Maths or Computer Science. This reveals that presence of these concepts in a sentence is a strong indicator of such sentence of being a definition in the encyclopedic genre.

Bag-of-non-Disambiguated Concepts

We extract the 500 most frequent text snippets that Babelfy did not disambiguate. The vector construction procedure is the same as in Bag-of-Concepts. In this case, we obtain results consistent with previous studies in that the pattern “is a” is the most frequent and hence a feature with high predictive power, followed by “is the”, “of a” and “is any”.

Semantic Features

We put forward a novel set of features stemming from the hypothesis that, in a definition, most concepts should be closely related, and hence should show higher semantic similarity than *distractor* sentences. For instance, in our working example “A bat is a mammal in the order Chiroptera”, the concepts *bat*, *mammal* and *Chiroptera* are closely related, and intuitively their corresponding vectors should be *more compact* and closer in the vector space, as opposed to one of its distractors in the WCL corpus: “This role explains environmental concerns when a bat is introduced in a new setting”. Here, concepts like *bat*, *to explain*, *environmental* or *setting* have a set of associated vectors *more sparsely distributed* in the vector space.

We build on this intuition to propose the following features:

- **AllSims** The sum of the SIM scores in Δ.
- **AvgSims** The average of the SIM scores in Δ.
- **AvgBiggestSubGraph** We can express our list of SIM scores as a non-directed cyclic graph, in which each node is a concept and each edge is weighted according to their SIM score. However, there are cases in which not all components of the graph are connected because one concept may be associated to two different lexicalizations depending on which concept it is disambiguated against. For instance, the concept for *mammal* in our working example may be lexicalized as *mammal* if disambiguated against *bat*, and as *mammalia* if disambiguated against *chiroptera.* This feature is the average of the cosine scores of the biggest connected subgraph generated from Δ[2]. Note that if the sentence graph is complete, **AvgSims** and **AvgBiggestSubGraph** yield the same score.
- **TopDegreeScore** First, we obtain the node with highest degree in the graph representation described above, i.e. the most connected node. Then, we compute the average SIM score over this node and its neighbours. We hypothesize that this measure should reward concepts whose disambiguation remains the same regardless of the concept they are disam-

[2] Graph operations performed in our experiments were done with the Python library NetworkX: https://networkx.github.io/

biguated against, which can be seen as another *semantic compactness* measure.

- **NumEdges** The number of edges of the graph described above. As the disambiguation options for a given concept increases, so will increase the number of edges of the graph representation. This is a feature aimed at capturing *non-definitional* sentences.
- **MaxScore and MinScore** The maximum and minimum SIM score among all the concept pairs in Δ. We hypothesize that in a definitional sentence, there will be at least one pair highly similar, the one between the defined term and the hypernym.

These features are used to perform a set of experiments with the machine learning toolkit WEKA (Witten and Frank, 2005). While many configurations and algorithms were tested, for brevity we report here the ones for the best performing experiment, based on Support Vector Machines.

3 Evaluation

Our approach (Our) shows competitive results, outperforming previous systems on the same dataset. We compare against three main competitors: (1) The WCL algorithm (WCL), which generalizes word-lattices over surface form and part-of-speech tags, hence producing word-class lattices (Navigli and Velardi, 2010); (2) A supervised machine-learning setting (BdC) in which syntactic dependencies are used to construct word representations in terms of their direct descendants (Boella et al., 2014); and (3) Another supervised approach (EspSag) also based on syntactic dependencies, but representing each sentence as a bag-of-dependency-subtrees (Espinosa-Anke and Saggion, 2014).

As is the case in all the systems described, performance is evaluated with the classic Precision, Recall and F-Score measures at sentence-level. Table 2 shows the performance of all systems.

We complement our experiments by evaluating the relevance of each individual feature from our feature set. To this end, we compute their Information Gain score, which measures the decrease in entropy when the feature is given vs. absent (Forman, 2003). The feature ranking provided in Table 3 shows the discriminative power of the features derived from SENSEMBED, reinforcing our claim that semantic information can be effectively applied to the DE task.

	Precision	**Recall**	**F-Score**
WCL	**98.8**	60.7	75.2
BdC	88.1	76.2	81.6
EspSag	85.9	85.3	85.4
Our	86.1	**86.0**	**86.0**

Table 2: Comparative results over the WCL dataset.

InfGain Score	**Feature**
“Contains:is_a”	0.19
AvgSims	0.13
AvgBiggestSubGraph	0.12
MaxScore	0.07
MinScore	0.06
TopDegreeScore	0.04
“Contains:is_an”	0.03
“Contains:bn00103785a”	0.02
NumEdges	0.01
AllSims	0.01

Table 3: Top 10 features according to their Information Gain score

4 Conclusions

Identifying definitional text snippets in free text is a task that can be integrated in more complex systems on ontology learning, dictionary or glossary construction, or for supporting terminological or eLearning applications. In this paper, we have described a supervised approach to DE that benefits substantially from introducing simple metrics derived from SENSEMBED, a sense-based vector representation of concepts and their lexicalizations. For future work, we would like to introduce features derived from the BabelNet graph, such as proximity, random walks or relation type; as well as adding additional vector comparison measures, e.g. the Tanimoto coefficient, used in (Iacobacci, Pilehvar, and Navigli, 2015).

References

Boella, Guido, Luigi Di Caro, Alice Ruggeri, and Livio Robaldo. 2014. Learning from syntax generalizations for automatic semantic annotation. *Journal of Intelligent Information Systems*, pages 1–16.

Cui, Hang, Min-Yen Kan, and Tat-Seng Chua. 2005. Generic soft pattern models for definitional question answering. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 384–391. ACM.

Espinosa-Anke, Luis and Horacio Saggion. 2014. Applying dependency relations to definition extraction. In *Natural Language Processing and Information Systems*. Springer, pages 63–74.

Forman, George. 2003. An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289–1305.

Iacobacci, Ignacio, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Enhancing word embeddings for semantic similarity and relatedness. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Beijing, China, July. Association for Computational Linguistics.

Jin, Yiping, Min-Yen Kan, Jun-Ping Ng, and Xiangnan He. 2013. Mining scientific terms and their definitions: A study of the ACL anthology. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 780–790, Seattle, Washington, USA, October. Association for Computational Linguistics.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751. Citeseer.

Moro, Andrea, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Muresan, A and Judith Klavans. 2002. A method for automatically building and evaluating dictionary resources. In *Proceedings of the Language Resources and Evaluation Conference (LREC*.

Navigli, Roberto and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Navigli, Roberto and Paola Velardi. 2006. Ontology enrichment through automatic semantic annotation of on-line glossaries. In *Managing Knowledge in a World of Networks*. Springer, pages 126–140.

Navigli, Roberto and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1318–1327, Stroudsburg, PA, USA. Association for Computational Linguistics.

Navigli, Roberto, Paola Velardi, and Juana María Ruiz-Martínez. 2010. An annotated dataset for extracting definitions and hypernyms from the web. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Park, Youngja, 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.

Rebeyrolle, Josette and Ludovic Tanguy. 2000. Repérage automatique de structures linguistiques en corpus : le cas des énoncés définitoires. *Cahiers de Grammaire*, 25:153–174.

Saggion, Horacio and Robert Gaizauskas. 2004. Mining on-line sources for definition knowledge. In *17th FLAIRS*, Miami Bearch, Florida.

Sarmento, Luís, Belinda Maia, Diana Santos, Ana Pinto, and Luís Cabral. 2006. Corpógrafo V3 From Terminological Aid to Semi-automatic Knowledge Engineering. In *5th International Conference on Language Resources and Evaluation (LREC'06)*, Geneva.

Snow, Rion, Daniel Jurafsky, and Andrew Y Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*.

Storrer, Angelika and Sandra Wellinghoff. 2006. Automated detection and annotation of term definitions in German text corpora. In *Conference on Language Resources and Evaluation (LREC)*.

Velardi, Paola, Stefano Faralli, and Roberto Navigli. 2013. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3):665–707.

Westerhout, Eline and Paola Monachesi. 2007. Extraction of Dutch definitory contexts for elearning purposes. *Proceedings of the Computational Linguistics in the Netherlands (CLIN 2007), Nijmegen, Netherlands*, pages 219–34.

Witten, Ian H and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann.

A Generative Model of Words and Relationships from Multiple Sources

Un modelo generativo para las palabras y relaciones de múltiples fuentes

Stephanie L. Hyland **Theofanis Karaletsos** **Gunnar Rätsch**

Computational Biology Center,
Memorial Sloan Kettering Cancer Center,
1275 York Avenue, New York, NY 10065
{hyland, karaletsos, raetsch}@cbio.mskcc.org

Abstract: We present a *generative model* of words as elements of a vector space, and semantic relationships as affine transformations on this space. By combining domain-specific structural knowledge and free text corpora, we obtain embeddings which are both semantically meaningful and useful for relationship prediction.
Keywords: Embedding, relationships, generative model, WordNet, Word2Vec

Resumen: Presentamos un *modelo generativo* para las palabras como elementos de un espacio vectorial, con relaciones semánticas como transformaciones afines en este espacio. Combinamos conocimiento previo de un dominio y obtenemos representaciones que son semánticamente significativa y útiles para prédecir las relaciones.
Palabras clave: Incorporación, relaciones, modelo generativo, WordNet, Word2Vec

1 *Introduction*

Finding vector representations for words generally requires a large corpus of sample sentences. These can be used to infer the *meaning* of words by examining the contexts in which they appear (the distributional hypothesis of language, see Sahlgren (2008)). However, these conditions may not extend easily to highly *specialised* language domains, such as medicine. In this case, the available corpora may be limited in size and expressivity. A doctor may never mention that anastrazole is a aromatase inhibitor (a type of cancer drug), for example, because they communicate sparsely, assuming the reader shares their expert knowledge of the intrinsic meaning of these words. In such cases, it is likely that even larger quantities of data are required, but the sensitive nature of such data makes this difficult to attain.

Fortunately, such specialised disciplines often create expressive *ontologies*, in the form of annotated relationships between terms. These may be semantic, such as dog is a type of animal, or derived from domain-specific knowledge, such as anaemia is an associated disease of leukaemia. (This is a relationship found in the medical ontology system UMLS, see Bodenreider (2004)). We observe that these relationships can be thought of as additional *contexts* from which co-occurrence statistics can be drawn; the set of diseases associated with leukaemia arguably share a common context, even if they may not co-occur in a sentence.

We would like to use this structured information to improve the quality of learned embeddings, as well as obtaining a representation for such relationships in this space. We do so by assuming that each relationship is an *operator* which transform words in a relationship-specific way. Intuitively, the action of these operators is to distort the embedding space, effectively allowing words to have multiple representations without requiring a full set of parameters for each relationship.

The intended effect on the underlying (untransformed) embedding is to encourage a solution which is more sensitive to the domain than would be achieved using only unstructured information. Since the posterior of an embedding procedure is in general highly multimodal, the weak constraints imposed by the structural information should encourage a more identifiable solution.

While we do not attempt to model higher-order language structure such as syntax, we consider a generative model in which the distance between terms in the embedded space describes the probability of their co-occurrence in a given context. By using such a generative approach, we learn the joint distribution of all term pairs in all contexts, and can ask questions such as *What is the probability of anaemia appearing in a sentence with imatinib* [1]*, given anaemia is a disease associated with leukaemia?* This introduces flexibility for subsequent analyses that was not available in previously proposed models.

Related Work Recent works have explored the use of relational data for learning word embeddings. Bordes et al. (2011) scored the similarity of entities (words) under a given relationship by their distance after transformation using pairs of relationship-specific matrices. Socher et al. (2013) describe a neural network architecture with a more complex scoring function, noting that the previous method does not allow for interactions between entities. Bordes et al. (2013) represent relationships as *translations*, motivated by the tree representation of hierarchical relationships, and observations that linear composition of entities appears to preserve semantic meaning (Mikolov et al. , 2013).

Similar in spirit to our work is Weston et al. (2013), where entities belonging to a structured database are identified in unstructured (free) text in order to obtain embeddings useful for relation prediction. However, they learn separate scoring functions for each data source. In our approach we consider a single energy function defining a distribution over the joint space of possible word pairs and relationships.

The `Word2Vec` model of Mikolov et al. (2013) is a special case of our model (when the only relationship is that of appears together in a sentence). In practice, `Word2Vec` uses a distinct objective function, since the full softmax is replaced by an approximation intended to avoid computing a normalising factor. As discussed in Section 2.1, we retain a probabilistic interpretation by approximating gradients of the partition function, at some computational cost.

The motivation for our work is similar in spirit to multitask and transfer learning (for instance, Caruana (1997), Evgeniou and Pontil (2004), or Widmer and Rätsch (2012)). In transfer learning one takes advantage of data related to a similar, typically supervised, learning task with the aim to improve the accuracy of a specific learning task. In our case, we have the unsupervised learning task of embedding words and relationships into a vector space and would like to use data from another task to improve the learned embeddings, here word co-occurence relationships. This may be understood as a case of *unsupervised transfer learning.*

2 *Mathematical Formulation*

We consider a probability distribution over triples (S, R, T) where S is the *source word* of the (possibly directional) *relationship* R and T is the *target word.* Following Mikolov et al. (2013), we learn two representations for each word: $\mathbf{c}_s$ represents word s when it appears as a *source*, and $\mathbf{v}_t$ for word t appearing as a *target.*[2] Relationships act by altering $\mathbf{c}_s$ through their action on the vector space ($\mathbf{c}_s \mapsto G_R\mathbf{c}_s$). By allowing G_R to be an arbitrary affine transformation, we combine the bilinear form of Socher et al. (2013) with translation operators of Bordes et al. (2013).

We use a Boltzmann probability distribution function,

$$\begin{aligned} P(S,R,T|\Theta) &= \tfrac{1}{Z(\Theta)} e^{-\mathcal{E}(S,R,T|\Theta)} \\ &= \frac{e^{-\mathcal{E}(S,R,T|\Theta)}}{\sum_{s,r,t} e^{-\mathcal{E}(s,r,t|\Theta)}} \end{aligned} \qquad (1)$$

and choose the energy function

$$\mathcal{E}(S,R,T|\Theta) = -\mathbf{v}_T \cdot G_R \mathbf{c}_S \qquad (2)$$

where $\Theta = \{\mathbf{c}_i, \mathbf{v}_j, G_r\}_{i,j,\in\text{vocabulary}}^{r\in relationships}$ is the set of parameters to be learned. By minimizing $\mathcal{E}(S,R,T|\Theta)$ by likelihood-maximization, we capture the desire that the representation of S be similar to that of T after it has been transformed by G_R.

2.1 Inference

We perform stochastic maximum-likelihood estimation using stochastic gradient descent. To avoid explicitly evaluating the gradient of the partition function, we use persistent contrastive divergence (PCD; Tieleman (2008)).

[1] Imatinib is a tyrosine-kinase inhibitor used in the treatment of chronic myelogenous leukaemia.

[2] Goldberg and Levy (2014) provide a motivation for using two representations for each word. We can extend this by observing that words with similar $\mathbf{v}$ representations share a *paradigmatic* relationship in that they may be exchangeable in sentences, but do not tend to co-occur. Conversely, words s and t with $\mathbf{c}_s \approx \mathbf{v}_t$ have a *syntagmatic* relationship and tend to co-occur (e.g., Sahlgren (2008)). Thus, we seek to enforce syntagmatic relationships and through transitivity obtain paradigmatic relationships of $\mathbf{v}$ vectors.

In traditional contrastive divergence, the gradient of the partition function is estimated using samples drawn from the model distribution seeded at the current training example (Hinton , 2002). However, many rounds of sampling may be required to obtain good samples. PCD retains a Markov chain of model samples across batches, assuming that the underlying distribution changes slowly. We use Gibbs sampling (between S, R, and T-type parameters) to obtain model samples.

In particular, we draw S from the conditional probability distribution:

$$P(S|r, t; \Theta) = \frac{e^{-\mathbf{v}_t \cdot G_r \mathbf{c}_S}}{\sum_{s'} e^{-\mathbf{v}_t \cdot G_r \mathbf{c}_{s'}}} \qquad (3)$$

and sequentially draw R and then T from analogous distributions. Thereby, we can estimate the gradient of $Z(\Theta)$ at the cost of these normalisation factors, which are linear in the size of the vocabulary. We use `Adam` (Kingma and Ba, 2014) to adapt learning rates and improve numerical stability.

2.2 Implementation

We provide the algorithm in Python, training data and other resources (`https://github.com/corcra/bf2`). Since most of its runtime takes place in vector operations, we are developing a GPU-optimized version using `Theano` (Bastien et al. , 2012).

For the experiments described in this manuscript, we used the following set of hyperparameters (using the notation from Kingma and Ba (2014)): $\alpha = 0{,}002$, $\lambda = 1 - 10^{-8}$, $\epsilon = 1 - 10^{-8}$, $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$. For PCD, we used 3 rounds of Gibbs sampling and 5 independent Markov chains. The batch size was 100. We used a vector dimension of $d = 100$.

3 Experiments

Data As structured data, we use the `WordNet` dataset described by Socher et al. (2013), available at `http://stanford.io/1IENOYH`. This contains 38,588 words and 11 relationships. Training data consists of true triples such as (feeling, has instance, pride). Since our model differs from others in several ways we tested it initially on this task to find good hyperparameters (see Section 2.2).

To incorporate *unstructured* data, we downloaded English Wikipedia (`https://dumps.wikimedia.org/`, August 2014) and extracted text using `WikiExtractor` (`http://bit.ly/1Imz1WJ`).

We aligned vocabularies between `WordNet` and Wikipedia by stripping sense IDs from the `WordNet` terms, and greedily identifying `WordNet` two-grams in the Wikipedia text. Stripping senses reduced the vocabulary to 33,330 words. We note that this procedure likely makes the prediction task more difficult, as each word receives only one representation.

Two words were considered in a sentence context if they appeared within a five word window. Only pairs for which both words appeared in the `WordNet` vocabulary were included. We drew from a pool of 112,581 training triples in `WordNet` with 11 relationships, and 8,206,304 triples from Wikipedia (heavily subsampled).

Adding Unstructured Data to a Relationship Prediction Task To test how unstructured text data may improve a prediction task when *structured* data is scarce, we augmented a subsampled set of triples from `WordNet` with 10,000 examples from Wikipedia and varied their weight κ in gradients during learning. The task is to predict whether or not a given triple (S, R, T) is a true example from `WordNet`. Following Socher et al. (2013), we use a validation set to find the (relationship-specific) *energy* threshold below which a triple is predicted as true.

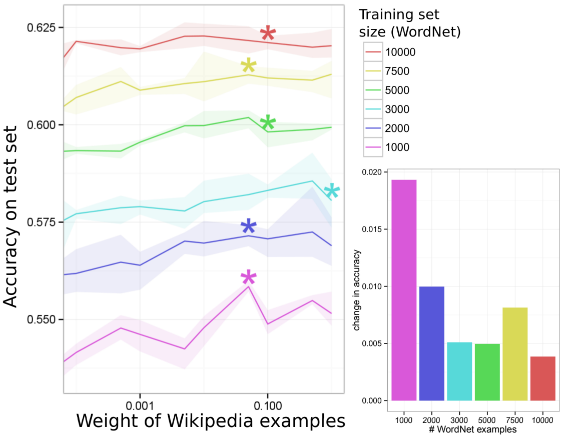


Fig. 1: Unstructured data helps relationship learning: unstructured data from Wikipedia is included with varying weight (x-axis) during training. Following Socher *et al.*, we predict if a triple (S, R, T) is true by using is energy as a score. A validation set is used to determine the threshold below which a triple is considered 'true'. The bar plot on shows the difference in accuracy between $\kappa = 0$ and $\kappa = \kappa^*$, where κ^* gave the highest accuracy on a validation set. The solid line denotes the average of three independent experimental runs; shaded areas show the range of results.

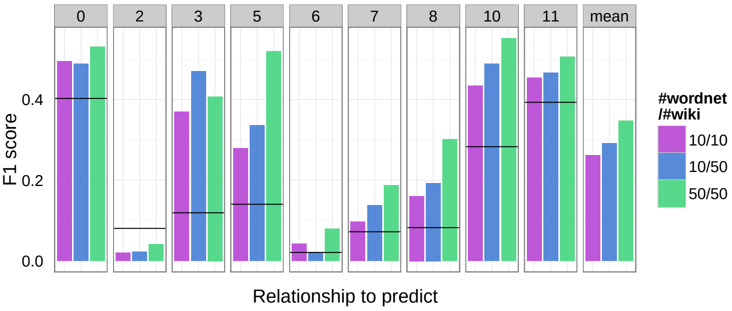


Fig. 2: Relationship data improves learned embeddings: We apply our algorithm on a scarce set of Wikipedia co-occurences (10k and 50k) with varying amounts of additional, unrelated relationship data (10k and 50k from `WordNet`). We test the quality of the embedding by measuring the accuracy on a task related to nine relationships (see main text; tested relationship was left out from training; we omitted relationships similar to, domain topic for technical reasons). Black lines denote results using vectors from `Word2Vec` trained on a Wikipedia-only dataset with 4,145,372 *sentences.*

Figure 1 shows accuracy on this task as κ and the amount of structured data vary. To find the improvement associated with *unstructured data*, we compared accuracy at $\kappa = 0$ with $\kappa = \kappa^*$ (where κ^* gave the highest accuracy on the validation set (marked with $*$)). We find that including free text data quite consistently improves the classification accuracy, particularly when the abundance of structured data is low.

Relationship Data for Improved Embeddings In this case, we assume *unstructured text data* is restricted, and vary the quantity of structured data. To evaluate the *untransformed* embeddings, we use them as the inputs to a supervised multi-class classifier. The task for a given (S, R, T) triple is to predict R given the vector formed by concatenating $\mathbf{c}_S$ and $\mathbf{v}_T$. We use a random forest classifier trained on the `WordNet` validation set using 5-fold cross-validation(Pedregosa et al. , 2011).

To avoid testing on the training data (since the embeddings are obtained using the `WordNet` training set), we perform this procedure once for each relationship (11 times), each time removing from the training data *all* triples containing that relationship. Figure 2 shows the F1 score of the multi-class classifier on the left-out relationship for different combinations of data set sizes. We see that for most relationships, including more unstructured data improves the embeddings (measured by performance on this task). Intriguingly, even data about *unrelated* relationships produces a performance increase, suggesting that including this structured information produces vectors that are semantically richer overall.

These results illustrate that embeddings learned from limited free text data can be improved by additional, unrelated relationship data.

4 Conclusion and Future Work

We have presented a *probabilistic generative model of words and relationships* between them. By optimising the parameters of this model through stochastic gradient descent, we obtain vector and matrix representations of these words and relationships respectively. To make inference tractable, we use *persistent contrastive divergence* with Gibbs sampling between entity types (S, R, T) to approximate gradients of the partition function. Our model uses an energy function which contains the idealised `Word2Vec` model as a special case. By augmenting the embedding space and considering relationships as arbitrary *affine* transformations, we combine benefits of previous models. In addition, our formulation as a generative model is distinct and allows a more flexible use.

Motivated by settings in which structured *or* unstructured data may be scarce (e.g., in the specialised healthcare domain), we illustrated how a model that combines both data sources can improve the quality of embeddings. While the presented analyses are preliminary, the *experimental results are very promising.*

Language models in general produce highly complex embeddings, and modelling choices may have a large impact. We are particularly interested in other choices of energy functions, specifically those that define *proper* distance metrics on the embedded space.

The intended future application of this model is *medical language processing.* We have a corpus of doctors text notes derived from electronic health records which we will augment with structured data from UMLS. We intend to perform feature-extraction on these notes using word representations and with the aim to improve performance by incorporating the domain knowledge encoded in medical ontologies.

References

[Bastien et al. 2012] Bastien, Frédéric, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. 2012. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

[Bodenreider 2004] Bodenreider, Olivier. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270.

[Bordes et al. 2013] Bordes, Antoine, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. En *Advances in Neural Information Processing Systems*, páginas 2787–2795.

[Bordes et al. 2011] Bordes, Antoine, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. En *Conference on Artificial Intelligence*, numero EPFL-CONF-192344.

[Caruana 1997] Caruana, Rich. 1997. Multitask Learning. *Machine Learning*, 28(1):41 – 75.

[Evgeniou and Pontil2004] Evgeniou, T. and M. Pontil. 2004. Regularized multi-task learning. En *International Conference on Knowledge Discovery and Data Mining*, páginas 109–117.

[Goldberg and Levy2014] Goldberg, Yoav and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722.*

[Hinton 2002] Hinton, Geoffrey E. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

[Kingma and Ba2014] Kingma, Diederik and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

[Mikolov et al. 2013] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. En *Advances in neural information processing systems*, páginas 3111–3119.

[Pedregosa et al. 2011] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Sahlgren 2008] Sahlgren, Magnus. 2008. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):33–53.

[Socher et al. 2013] Socher, Richard, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. En *Advances in Neural Information Processing Systems*, páginas 926–934.

[Tieleman 2008] Tieleman, Tijmen. 2008. Training restricted boltzmann machines using approximations to the likelihood gradient. En *Proceedings of the 25th international conference on Machine learning*, páginas 1064–1071. ACM.

[Weston et al. 2013] Weston, Jason, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. En *Conference on Empirical Methods in Natural Language Processing*, páginas 1366–1371.

[Widmer and Rätsch2012] Widmer, C. and G. Rätsch. 2012. Multitask Learning in Computational Biology. *JMLR W&CP. ICML 2011 Unsupervised and Transfer Learning Workshop.*, 27:207–216.

Proposing distributional semantics as a tool for medical vocabulary expansion

Mofizur Rahman and Lars Asker
DSV, Stockholm University
Postbox 7003, 164 07 Kista
{mora0069,asker}@dsv.su.se

Maria Skeppstedt
Linnaeus University
351 95 Växjö
mariask@dsv.su.se

Abstract: A tool that extends a given vocabulary by automatically extracting new term candidates from a corpus could facilitate vocabulary expansion, as well as ensure that extracted terms correspond to those actually used in a specific text genre. We here propose a user interface for such a tool, and evaluate the feasibility of using Random Indexing for positioning new term candidates in a given taxonomy.

Keywords: Vocabulary expansion, Random Indexing, Medical Vocabulary

1 Introduction

Extensive vocabularies are essential for automatic processing of medical texts, and there are a number of such medical vocabularies, many of them ordered in a taxonomic structure (Bodenreider, 2004).

In addition to being expensive to develop, however, terms included in medical vocabularies do not always correspond to those actually used in medical texts. This can lead to low performance for automatic processing tasks, in particular for smaller languages, for which the vocabularies often consist of translated, reduced versions of the original resources (Skeppstedt, Kvist, and Dalianis, 2012). A tool that provides candidates for new terms to include in the vocabulary by extracting terms from a corpus could, however, facilitate vocabulary expansion. In addition, this approach would ensure that extracted candidates correspond to terms actually used in the genre of the corpus. We here propose a user interface for such a tool, and perform a feasibility evaluation of the underlying functionality for providing term candidates.

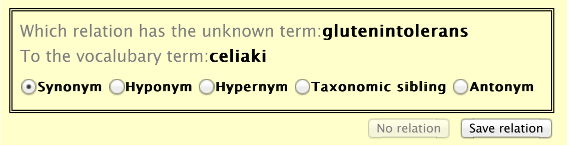
2 User interface for a term extraction tool

Given the task of expanding a vocabulary from a medical corpus, a system should: (1) extract terms that are typical for the genre (i.e. medical terms) and not yet included in the vocabulary, (2) for each such term, present the user with similar terms that are already included in the vocabulary. The user will then have the opportunity to indicate the type of semantic relation (synonym, antonym, hyper/hyponym or taxonomic siblings). Thereby, the system can correctly position the new term in the taxonomic structure of an existing vocabulary. A prototype for such an interface is shown in Figure 1.

3 Provide term candidates

The first task, extraction of genre specific terms, could be carried out by comparing tf-idf (term frequency – inverse document frequency) for medical and non-medical texts (Robertson, 2004).

For the second task, suggesting similar terms, distributional semantics can be applied. In contrast to vocabulary extraction approaches that rely on terms being explicitly defined or explained in the text (Hearst, 1992; Neelakantan and Collins, 2014), distributional semantics methods are able to provide similar terms candidates for any corpus term. Examples of such techniques are clustering of terms with similar neighbours (Lin, 1998), or continuous distributional semantics (word embedding) representations such as word2vec (Mikolov et al., 2013) and the technique employed here, i.e., Random In-



Which relation has the unknown term:**glutenintolerans**

To the vocalubary term:**celiaki**

● **Synonym** ○ **Hyponym** ○ **Hypernym** ○ **Taxonomic sibling** ○ **Antonym**

Figure 1: Positioning the new candidate term *glutenintolerans* in an existing vocabulary.

dexing (RI) (Sahlgren, Holst, and Kanerva, 2008).

Henriksson et al. (2014) used RI for medical vocabulary expansion, for the task of suggesting a correct synonym among ten term candidates. In the present work, with the purpose of aiming for a higher recall, we evaluate the performance of RI using the same medical corpus, but allowing the model to suggest more term candidates. This corpus is the freely available subset (years 1996–2005, 21,447,900 tokens) of *Läkartidningen* (Journal of the Swedish Medical Association) (Kokkinakis, 2012). The considerably smaller corpus size than commonly used for distributional semantics is realistic, due to the limited availability of large medical corpora for smaller languages.

A 1000-dimensional RI space was created, using a context window of two preceding and two following words and giving double weight to the direct neighbours of the target word. Context windows were not allowed to cross sentence boundaries.

We used 93 synonym pairs of one-token terms from Swedish MeSH (KI, 2012) as evaluation data (all occurring more than 50 times in the corpus). An automatic evaluation was carried out by searching for the top n most similar terms in the RI model to one of the terms in each synonym pair. We, thereafter, measured the recall for retrieving the other term in the pair among these n candidates. The results for different cut-offs of n are shown in Figure 2. Using the top 50 term candidates, instead of the top 10, improves recall with 13 percentage points, clearly showing the benefit of using a longer candidate list.

The low precision is likely to be improved in the scenario of the suggested tool, since the candidate lists would only contain terms already included in existing vocabularies. Candidates that are distantly positioned in the semantic space could also be removed, and the taxonomic structure of existing vocabularies could be leveraged, e.g., by removing synonymous terms from the candidate list.

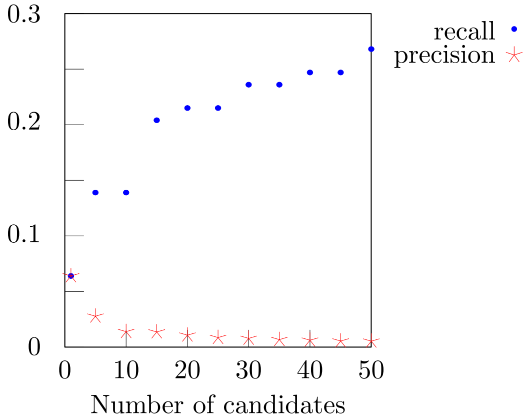


Figure 2: Recall and precision for different cut-offs in terms of number of candidates.

The recall is lower than simliar, previous studies for Swedish conducted on larger corpora (Henriksson et al., 2013; Henriksson et al., 2014), but those larger corpora were obtained by using large resources of clinical texts, which are only rarely made available for research. Approaches by Henriksson et al. (2014) to improve results by creating ensembles of semantic spaces are, however, more generally applicable.

4 Conclusion

A prototype interface for vocabulary expansion of medical vocabularies was constructed, and the underlying functionality for providing term candidates for such a tool was outlined. The feasibility evaluation of this functionality showed the benefit of a longer candidate list: using the top 50 term candidates, instead of the previously used top 10, improved recall with 13 percentage points.

Future work includes implementing the entire proposed functionality for providing term candidates, and integrating this with the proposed user interface and all available Swedish medical vocabularies. We also intend to further develop the interface, e.g., by a graphical presentation of the taxonomic structure of existing vocabularies and the proposed position of each new term candidate.

References

Bodenreider, Olivier. 2004. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucl. Acids Res.*, 32(suppl 1):D267–270, January.

Hearst, Marti. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING 1992*, pages 539–545.

Henriksson, Aron, Hans Moen, Maria Skeppstedt, Vidas Daudaravičius, and Martin Duneld. 2014. Synonym extraction and abbreviation expansion with ensembles of semantic spaces. *J Biomed Semantics*, 5(1):6.

Henriksson, Aron, Maria Skeppstedt, Maria Kvist, Martin Duneld, and Mike Conway. 2013. Corpus-driven terminology development: Populating swedish snomed ct with synonyms extracted from electronic health records. In *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing*, pages 36–44, Sofia, Bulgaria, August. Association for Computational Linguistics.

KI. 2012. Karolinska Institutet: Hur man använder den svenska MeSHen (In Swedish, translated as: How to use the Swedish MeSH). http://mesh.kib.ki.se/swemesh/manual_se.html. Accessed 2012-03-10.

Kokkinakis, Dimitrios. 2012. The journal of the Swedish medical association - a corpus resource for biomedical text mining in Swedish. In *Proceedings of BioTxtM*.

Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, pages 768–774, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Neelakantan, Arvind and Michael Collins. 2014. Learning dictionaries for named entity recognition using minimal supervision. In Gosse Bouma and Yannick Parmentier 0001, editors, *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 452–461. The Association for Computer Linguistics.

Robertson, Stephen. 2004. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60:2004.

Sahlgren, Magnus, Anders Holst, and Pentti Kanerva. 2008. Permutations as a means to encode order in word space. In *Proceedings CogSci 2008*.

Skeppstedt, Maria, Maria Kvist, and Hercules Dalianis. 2012. Rule-based entity recognition and coverage of SNOMED CT in Swedish clinical text. In *Proceedings of LREC'12*.

Espacios vectoriales complejos: un estudio exploratorio sobre la inclusión del orden de las palabras en los modelos vectoriales

Complex vector spaces: an exploratory study on the incorporation of word-order information in vector space models

Rafael E. Banchs
Institute for Infocomm Research
1 Fusionopolis Way #21-01, Singapore 138632
rembanchs@i2r.a-star.edu.sg

Resumen: En este trabajo de investigación exploramos el uso del álgebra compleja para generar espacios vectoriales sensibles al orden de las palabras, con el fin de modelar construcciones gramaticales cortas como *n*-gramas, frases y oraciones. A diferencia de otras aproximaciones tradicionales basadas en espacios vectoriales de números reales, como los modelos basados en bolsas-de-palabras, en nuestra aproximación proponemos el uso de espacios vectoriales de números complejos para modelar simultáneamente la información de la ocurrencia y del orden de las palabras. Adicionalmente mostramos cómo es posible generar representaciones de baja dimensionalidad para este tipo de modelos y exploramos algunas de sus propiedades básicas tanto con secuencias artificiales de símbolos como con muestras reales de lenguaje natural.
Palabras clave: Representaciones de baja dimensionalidad, Modelos basados en Espacios Vectoriales, Aritmética Compleja, Información sobre el Orden de las Palabras, Reducción de la Dimensionalidad.

Abstract: In this research work we explore the use of complex algebra to generate word-order aware vector spaces able to model short language constructs, such as *n*-grams, phrases and sentences. Different from the traditional bag-of-word model approach, which mainly models word occurrences by means of a real-valued vector space, in the proposed framework, we use complex-valued representations to account for both word-occurrence and word-order information. In this paper we introduce the proposed approach and show how reduced-dimensionality embeddings can be generated for this type of models. We also explore the basic properties of the resulting embeddings with both artificial sequences and real natural language data.
Keywords: Embeddings, Vector Space Models, Complex Arithmetic, Word Order Information, Dimensionality Reduction.

1 Introduction

The use of vector spaces and continuous space embeddings has been used in both computational linguistics and natural language processing for more than twenty years (Spärk Jones 1972, Salton et al. 1975, Deerwester et al. 1990, Baroni and Lenci 2010, Turney and Pantel 2010). Theoretically supported by the distributional hypothesis (Firth 1957, Harris 1970, Sahlgren 2006), this framework has been proven to be useful for modeling different aspects of the linguistic phenomena, with a special emphasis on semantics (Turney and Pantel 2010).

One of the main limitations of the original approach is that word-order information is not taken into account. One major consequence of this limitation is that good models can be produced for documents and for words, but not for intermediate level units such as sentences or phrases, in which word-order information plays a very important role.

Recent research work targeting the problem of including language structure information on vector space and continuous space models includes the use of additive and multiplicative models (Mitchell and Lapata 2008), circular convolution models (Jones and Mewhort 2007), random permutation models (Sahlgren et al. 2008), recursive matrix vector spaces (Socher et al. 2012) and recurrent neural networks (Mikolov et al. 2013), among others (Erk and Pado 2008, Recchia et al. 2010, Baroni and Zamparelli 2010).

In this work, we explore the use of complex algebra to implement a word-order aware vector space model able to represent short language constructs, such as *n*-grams, phrases and sentences. Different from the traditional bag-of-word model approach, which mainly models word occurrences in a real-valued vector space, in the proposed framework, we use a complex-valued vector space to account for both word-occurrence and word-order information.

Furthermore, we also show how real-valued continuous space embeddings can be derived from the proposed complex-valued vector space representations, and how these resulting embeddings are sensitive to the structural properties of language.

The rest of the paper is organized as follows. First, in section 2, we introduce the proposed approach. Then, in section 3, we illustrate how continuous space embeddings can be generated for these models and explore some of their basic properties for the case of artificial sequences of symbols. Later, in section 4, we show with actual natural language data how the embeddings resulting from complex vector space representations are more sensitive to language structure than those resulting from conventional real-valued vector spaces. Finally, in section 5, we present the main conclusions of this exploratory work and propose some future research avenues in this area.

2 Complex vector spaces

In this section we introduce the complex vector space model framework for language representation, which allows for simultaneously modelling word-occurrence and word-order information into a single vector space representation.

First, let us formalize the concepts of amplitude and phase for a word w_x into a given segment of text $S = \{w_1,w_2,w_3 \ldots w_n\}$. By amplitude (*Am*), we will refer to the number of times the word w_x occurs into the given segment of text. Notice that this concept is completely equivalent to the conventional concept of term frequency. On the other hand, by phase (*Ph*), we will refer to the relative average position of such a word w_x within the given segment of text. According to these definitions we can write:

$$Am\{w_x\} = \Sigma_{i=1\ldots n}\, Ind(w_x,w_i)$$

$$Ph\{w_x\} = 1/Am\{w_x\}\ \Sigma_{i=1\ldots n}\, Idx(w_x,w_i)/(n+1)$$

where $Ind(w_x,w_i)$ is the indicator function, which is 1 if $w_x = w_i$ and 0 otherwise; $Idx(\mathrm{w_x},\mathrm{w_i})$ is the index-indicator function, which is equal to i if $w_x = w_i$ and 0 otherwise; n is the total number of words in the given text segment S; and x ranges from 1 to the vocabulary size K.

Notice from the previous two definitions that the amplitude of a given word w_x corresponds to the count of the number of times it occurs within the text segment S; while its phase corresponds to its normalized average position inside text segment S. Normalized positions range from $1/(n+1)$ to $n/(n+1)$ for the first and last words in text segment S, respectively.

Now, we can define the proposed complex vector space representation of a given text segment S as a vector $\boldsymbol{V_S}$ of size K (where K is the size of the vocabulary), in which each of its elements x is computed as follows:

$$V_S[x] = Am\{\mathrm{w}_x\}\ \exp(\alpha\pi j\ (Ph\{\mathrm{w}_x\}\text{-}½))$$

where j is the unitary imaginary number, i.e. $sqrt(-1)$; and α is the phase-emphasizing factor, which can range between 0 and 2. For $\alpha = 0$ the proposed complex-valued space representation reduces to the conventional real-valued one. The maximum value of 2 is constrained by the periodicity of the complex exponential.

To illustrate how this space representation improves word-order awareness with respect to the traditional bag-of-word vector space model we will consider a toy example, but first let us define an extended version of the cosine similarity metric that can be used in complex spaces.

The main idea of cosine similarity is to use the cosine of the angle between two vectors as a measure of similarity or proximity. In real valued spaces, the cosine of the angle between two vectors is computed as follows:

$$sim(v_1,v_2) = < v_1,v_2 > / \,||v_1||\, / \,||v_2||$$

where $<v_1,v_2>$ is the internal product of the two vectors and $||v||$ is the quadratic norm operator. In complex space, both the internal product and the quadratic norm can be also computed, but the resulting similarity is not necessarily a real-valued score. To ensure a real-valued score for all possible similarities in complex space, we defined the following similarity score:

$$cplxsim = \|sim\| \cos(\phi(sim))$$

where $\|sim\|$ and $\phi(sim)$ are the amplitude and phase of the resulting similarity complex-valued score. Notice that this proposition is consistent with the similarity score for real-valued vector spaces, as for two real-valued vectors v_1 and v_2 it follows that $cplxsim(v_1,v_2) = sim(v_1,v_2)$.

Let us now consider the toy example mentioned before to illustrate the advantages of the proposed framework with respect to modeling word position information. Consider, for instance, the *3*-symbol vocabulary $\{a,b,c\}$ and the six different bigrams that can be constructed with it $\{ab,ac,ba,bc,ca,cb\}$. Conventional bag-of-word vector representations can only discriminate between bigrams containing different symbols:

$$V_{ab} = V_{ba} = [\ 1\ 1\ 0\ ]$$

$$V_{ac} = V_{ca} = [\ 1\ 0\ 1\ ]$$

$$V_{bc} = V_{cb} = [\ 0\ 1\ 1\ ]$$

Consequently, similarities between vectors containing the same symbols will be *1*; and similarities between vectors sharing one symbol will be *0.5*, as shown in the Table 1.

	ac	*ba*	*bc*	*ca*	*cb*
ab	*0.5*	*1*	*0.5*	*0.5*	*0.5*
ac	-	*0.5*	*0.5*	*1*	*0.5*
ba	-	-	*0.5*	*0.5*	*0.5*
bc	-	-	-	*0.5*	*1*
ca	-	-	-	-	*0.5*

Table 1: Cosine similarities between bigrams of a *3*-symbol vocabulary computed on real-valued vector space representations

By considering now the proposed complex-valued vector space, vector representations for the six bigrams will be as follows:

$$V_{ab} = [\ exp(-1/6\ \alpha\pi j)\ \ exp(1/6\ \alpha\pi j)\ \ 0\ ]$$

$$V_{ba} = [\ exp(1/6\ \alpha\pi j)\ \ exp(-1/6\ \alpha\pi j)\ \ 0\ ]$$

$$V_{ac} = [\ exp(-1/6\ \alpha\pi j)\ \ 0\ \ exp(1/6\ \alpha\pi j)\ ]$$

$$V_{ca} = [\ exp(1/6\ \alpha\pi j)\ \ 0\ \ exp(-1/6\ \alpha\pi j)\ ]$$

$$V_{bc} = [\ 0\ \ exp(-1/6\ \alpha\pi j)\ \ exp(1/6\ \alpha\pi j)\ ]$$

$$V_{cb} = [\ 0\ \ exp(1/6\ \alpha\pi j)\ \ exp(-1/6\ \alpha\pi j)\ ]$$

By setting the phase-emphasizing factor α to *1*, the resulting similarities between bigrams are as shown in Table 2. Notice how the model now penalizes symbol-order mismatches. More specifically, when the bigrams share the same symbols but in inverted positions, the similarity score is *0.5*, such as in the case when only one symbol is matched in the correct position. On the other hand, when only one symbol is shared by the bigrams and it is in the wrong position, the similarity score is *0.25*.

	ac	*ba*	*bc*	*ca*	*cb*
ab	*0.5*	*0.5*	*0.25*	*0.25*	*0.5*
ac	-	*0.25*	*0.5*	*0.5*	*0.25*
ba	-	-	*0.5*	*0.5*	*0.25*
bc	-	-	-	*0.25*	*0.5*
ca	-	-	-	-	*0.5*

Table 2: Cosine similarities between bigrams of a *3*-symbol vocabulary computed on a complex-valued vector space model with α = *1*.

If we want to further discriminate symbol occurrence from symbol order without penalizing too much position mismatch, we can just set the phase-emphasizing factor α to a smaller value. Table 3 presents the resulting similarities between bigrams when setting α to *½*.

	ac	*ba*	*bc*	*ca*	*cb*
ab	*0.5*	*0.87*	*0.43*	*0.43*	*0.5*
ac	-	*0.43*	*0.5*	*0.87*	*0.43*
ba	-	-	*0.5*	*0.5*	*0.43*
bc	-	-	-	*0.43*	*0.87*
ca	-	-	-	-	*0.5*

Table 3: Cosine similarities between bigrams of a *3*-symbol vocabulary computed on a complex-valued vector space model with α = *½*.

Notice from Table 3 how the model is able to better discriminate between the case of two shared symbols in inverted positions and the case of just one shared symbol in the correct position. Now, bigrams that share the two symbols but in inverted positions receive an score of *0.87*, while those that share only one symbol in the correct position still get a score of *0.5*; and those that share only one symbol but in the wrong position receive a score of *0.43*.

3 Space dimensionality reduction

In this section we show how low-dimensional embeddings can be generated for the proposed complex-valued vector space models. We also explore the basic properties of the resulting embeddings for artificial sequences of symbols under different conditions.

For embedding generation and result visualization we use MDS with Sammon's non-linear mapping criterion (Cox and Cox 2001). For illustrative purposes, we use small data collections as this also allows for a clear visualization of results in two-dimensional maps.

In our first example, we compute *2*-dimensional embeddings for all possible trigrams that can be generated with the *3*-symbol vocabulary $\{a,b,c\}$ while varying the phase-emphasizing factor α. These results are depicted in Figure 1.

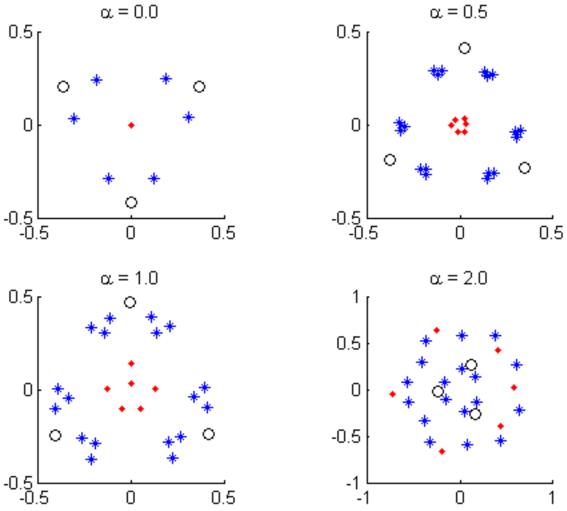


Figure 1: MDS-generated *2*-dimensional embedding for all trigrams in a *3*-symbol vocabulary with different values of α

As seen from Figure 1, three different types of trigrams *xyz* can be identified: those containing one symbol, i.e. with $x = y = z$ (which are depicted as circles in the figure); those containing two symbols, i.e. with either $x = y \neq z$, $x \neq y = z$ or $x = z \neq y$ (which are depicted as stars); and those containing three symbols, i.e. $x \neq y \neq z$ (which are depicted as dots).

In the case of $\alpha = 0$, no word-order information is encoded into the model, as the model reduces to the real-valued vector space model. As seen from the figure (upper-left panel), all *18* two-symbol trigrams are conflated into six representations as the model is not able to distinguish among trigrams of the form *xxy*, *xyx* and *yxx*. Similarly, all *6* three-symbol trigrams are conflated into a single point in the center of the map, as the model is not able to distinguish among the permutations of the three symbols.

On the other hand, when $\alpha > 0$ is used, the model becomes able to discriminate among all different trigrams as the word-order is accounted for by the phase component of the model. Moreover, nice clusters for each different types of trigrams can be distinguished when the value of α is moderately small (see upper-right and lower-left panels of the figure). However, when α is set to its maximum value of *2*, the clusters got so dispersed that different types of trigrams get mixed up among them.

In general, the total number of *n*-grams that are possible for a given vocabulary of size *K* is given by:

$$|\, n\text{-grams} \,| = K^{n}$$

where $|\cdot|$ represents the cardinality operator.

From all these *n*-grams, the number of those containing *n* different symbols (i.e. the ones corresponding to dots in Figure 1) is given by:

$$|\, n\text{-grams}_{\text{ without repetitions}} \,| = C(K,n)\ P(n)$$

where *C(K,n)* are all the possible combinations of the *K* elements in the vocabulary in groups of size *n*, and *P(n)* are the permutations of *n*. More specifically:

$$C(K,n) = K! \,/\, (n!\ (K\text{-}n)!)$$

$$P(n) = n!$$

Notice that from all these *n*-grams without repeated symbols, a real-valued vector space model can only distinguish the combinations as all the permutations will be conflated into a single representation. On the other hand, the complex-valued vector space model is able to discriminate all of them.

In general, when moderately small values of α are used, *n*-grams of this type will appear in the embedding as *C(K,n)* clusters of *P(n)* elements each. For instance, in Figure 1, only one cluster can be seen as *C(3,3) = 1*.

Regarding the number of *n*-grams containing only one symbol (the ones corresponding to circles in Figure 1), it is determined by the size of the vocabulary *K*, and they tend to appear in the embedding as polarized and isolated points.

Finally, the number of *n*-grams repeating one of its symbols or more (those corresponding to stars in Figure 1) is given by the difference $K^{n} - (C(K,n)\ P(n) + K\ )$, and they will tend to cluster according to number of repetitions and shared symbols they contain.

Figure 2 illustrates the case of all *4*-grams without repeated symbols that can be produced with a vocabulary of size *6*. The value of α used is ½. As seen from the example, in the resulting embedding, all the *4*-grams appear organized in *C(6,4) = 15* clusters of *P(4) = 24* elements each.

Notice that only a subset of *360 4*-grams, out of all the *1296* possibilities (6^4), is presented in the figure. All *4*-grams with repeated symbols have been excluded, as the visualization of such a large set of samples in a *2*-dimensional embedding becomes very clumsy.

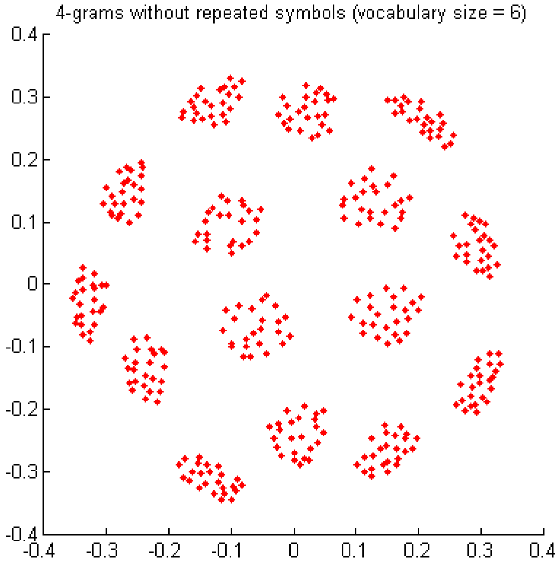


Figure 3: MDS-generated *2*-dimensional embedding for all *4*-grams without repeated symbols from a *6*-symbol vocabulary

One interesting by-product of the proposed complex-valued vector space model is that, in general, the original sequences of symbols can be recovered from their corresponding vector representations. However, this is not always the case for *n*-grams with repeated symbols.

Consider for instance the two *4*-grams *abba* and *baab*. As the proposed word phase component averages the relative positions of the symbols, the phase values for both *a* and *b* will be exactly the same in the two cases described above. Moreover, as both symbols appear twice in both *4*-grams, the word amplitude components will be also the same. Indeed, the amplitude and phase components of *a* and *b* for *abba* and *baab* will be as follows:

$$Am\{a\} = Am\{b\} = 2$$

$$Ph\{a\} = Ph\{b\} = ½$$

which means that complex-valued vector representations for both *abba* and *baab* are conflated into the same vector.

This constitutes indeed a more general problem, as much more complex symmetry issues are expected to arise when lager *n*-grams and vocabulary sizes are considered. Although this constitutes an important limitation of the proposed framework; the truth is that such special cases of word symmetry have low probability of occurring in natural language constructions. However, further research in this direction is needed in order to overcome this limitation.

4 Examples with natural language

In this section we will apply the proposed complex-valued vector models to actual sequences of words occurring in natural language. The main goal of the experiments in this section is to show that the proposed model is more sensitive to language structures than the conventional real-valued vector space model.

At this point, it is important to recall that real-valued vector spaces have been successfully used in natural language processing for several years already. Indeed, some empirical research has suggested that word frequency by itself accounts for about *80%* of the semantic information conveyed in language, while word-position information is believed to account only for the remaining *20%* (Landauer 2002).

This said, we do not expect extraordinary differences between the proposed framework and the real-valued one. However, we believe that the additional discrimination power provided by the phase-emphasizing factor and the possibility of recovering original text structures from its vector representations are, in general, interesting properties worth to be studied.

In this section we consider a small collection of sub-sentence units in the legal domain. The samples have been extracted from legal texts and, differently from plain *n*-grams, the segments have been extracted by using punctuation as the segmentation criterion. More specifically, text fragments delimited by punctuation marks such as . , ; : (- and so on, were extracted; and only segments with lengths between *4* and *7* words have been retained. Table 4 summarizes the distribution of text segments according to their length in number of words.

Words	4	5	6	7
Segments	80	88	70	85

Table 4: Word length distributions for the considered small collection of legal texts

By using the same procedure described in the previous section, we first constructed complex-valued vector representations for all the text segments and, then, computed MDS-based low-dimensional embeddings. In order to measure the ability of discriminating natural language structures, we paid attention to specific subsets of data samples within the embedding, which have been defined according to the relative positions of function words pairs. For instance, given the set of samples containing both words *to* and *be*, we defined two subsets according to the difference between the phase components of both words. Then, a given sample belongs to subset-1 if $Ph\{to\} < Ph\{be\}$ and to subset-2 if $Ph\{to\} > Ph\{be\}$.

According to this, text segments such as '*the procedure to be followed before it*' or '*in order to be valid*' are assigned to subset-1; while text segments such as '*shall be subordinated to the general interest*' or '*might be contrary to the constitution*' are assigned to subset-2.

Afterwards, we computed the inter-cluster distance between both subsets in the constructed embedding. In this sense, we expect that the more structure-aware a given embedding is, the larger inter-cluster distance should be observed. In order to compare the structure discrimination power of the embeddings, we constructed two: one by setting the phase-emphasizing factor α to *0* (real-valued vector space) and the other by setting α to *1* (complex-valued vector space).

We computed the inter-cluster distances for different structural pairs of subsets, more specifically: *be-to* vs. *to-be*, *of-the* vs. *the-of*, *for-the* vs. *the-for*, *and-the* vs. *the-and, of-and* vs. *and-of*, and *by-the* vs. *the-by*. Figure 4 presents the results of the comparative analysis for all proposed structural sets over the two constructed embeddings. All the inter-cluster distances are computed as the Euclidean norm of the difference vector between centroids of both subsets.

As seen from the figure, with the exception of the small difference observed for the structural group containing *of* and *the*, significant differences can be appreciated between inter-cluster distances computed in both embeddings. In general, inter-cluster distances are higher in the embedding generated from the complex-valued vector representations.

In order to discard the possibility of inter-cluster distances increasing due to overall set expansion, we conducted a control experiment. In this sense, we also computed the average intra-cluster distance for each complete set in both embeddings. The average intra-cluster distance is computed as the average Euclidean norm of vector differences among all vector pairs in each set. We then computed the ratio between the increment rate of the overall set (the ratio between intra-cluster distances in both embeddings) and the discrimination rate between subsets (the ratio between inter-cluster distances in both embeddings).

Again, with the exception of the structural group containing *of* and *the* (for which the resulting score was *1.06*), all scores where below *1*, more specifically, below *0.85*. This means that, in general, the rate of average intra-cluster distance expansion is smaller than the rate of inter-cluster distance increase. This suggests that, effectively, the embedding derived from complex-valued vector representations discriminates language structures better than the one derived from real-valued vector representations.

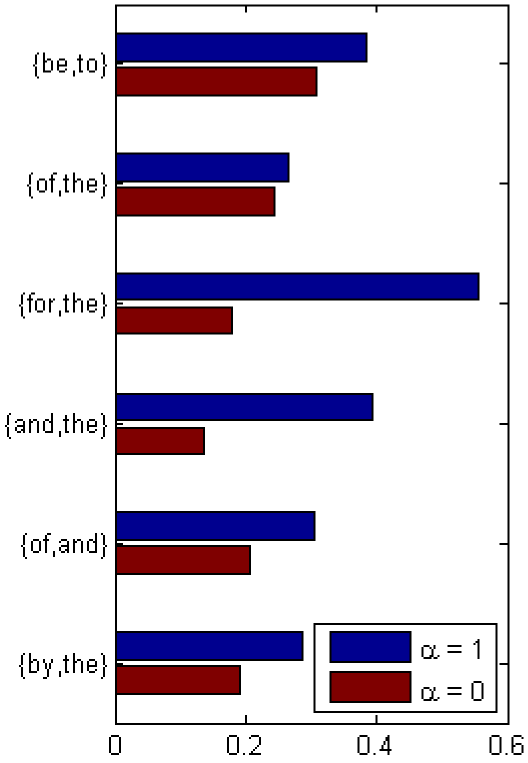


Figure 4: Inter-cluster distances between subsets of structural groups computed over embeddings of real and complex-valued vector spaces ($\alpha = 0$ and $\alpha = 1$, respectively)

5 Conclusions and future work

In this work we have proposed a word-order aware vector space model for representing short language constructs, such as *n*-grams, phrases and sentences. Different from traditional bag-of-word model approaches that model word occurrences in real-valued vector spaces, we use a complex-valued vector space to account for both word occurrence and order information.

We have explored the main properties of the proposed framework and we have shown its suitability for building low-dimensional embeddings that are more sensitive to structural differences in both natural language and artificial sequences of symbols in general.

One interesting by-product of the proposed framework is that, in general and with the exception of some symmetrical structures, the original sequences of words or symbols can be recovered from their corresponding vector representations.

As future work we plan to work in two directions. First, we will explore the advantages, if any, of using the proposed framework in practical natural language processing applications such as information retrieval, question answering and the like.

Second, we plan to continue exploring different parameterizations of the proposed model and to evaluate their potential applications in other fields. In particular we are interested in the problem of making the model fully invertible, regardless of the symmetry exhibited by the represented sequence of symbols.

References

M. Baroni, A. Lenci. 2010. Distributional Memory: A General Framework for Corpus-Based Semantics, Computational Linguistics, 36(4)

M. Baroni, R. Zamparelli. 2010. Nous are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space, in proceedings of EMNLP 2010

M.F. Cox, M.A.A. Cox. 2001. Multidimensional Scaling. Chapman and Hall, Boca Raton.

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman. 1990. Indexing by latent semantic analysis, Journal of the American Society for Inf. Science, 41: 391-407

K. Erk, S. Pado. 2008. A Structured Vector Space Model for Word Meaning in Context, in Proceedings of EMNLP 2008, pp. 897–906

J.R. Firth. 1957. A synopsis of linguistic theory 1930-1955, Studies in linguistic analysis, 51: 1-31

Z. Harris. 1970. Distributional Structure, in Papers in Structural and Transformational Linguistics, pp.775-794

M.N. Jones, D.J.K. Mewhort. 2007. Representing word meaning and order information in a composite holographic lexicon, Psychological Review, 114:1-37

T.K. Landauer. 2002. On the computational basis of learning and cognition: Arguments from LSA, in Ross B.H. (ed.) The Psychology of Learning and Motivation: Advances in Research and Theory, 41, pp. 43-84

T. Mikolov, W.T. Yih, G. Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations, NAACL-HLT 2013

J. Mitchell, M. Lapata. 2008. Vector-based models of semantic composition, in Proceedings of ACL–HLT 2008, pp. 236-244

G.L. Recchia, M.N. Jones, M. Sahlgren, P. Kanerva. 2010. Encoding sequential information in vector space models of semantics: Comparing holographic reduced representations and random permutations, in Proceedings of the 32nd Annual Cognitive Science Society

M. Sahlgren. 2006. The word-space model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. PhD Dissertation, Department of Linguistics, Stockholm University

M. Sahlgren, A. Holst, P. Kanerva. 2008. Permutations as a means to encode order in word space, in Proceedings of the 30th Annual Conference of the Cognitive Science Society, pp. 1300-1305

G. Salton, A. Wong, C.S. Yang. 1975. A vector space model for information retrieval. Communications of ACM 18(11):613-620.

R. Socher, B. Huval, C.D. Manning, A.Y. Ng. 2012. Semantic Compositionality through Recursive Matrix-Vector Spaces, in Proceedings of EMNLP-CoNLL 2012, pp. 1201-1211

K. Spärk Jones. 1972. A statistical interprettation of term specificity and its application in retrieval. Journal of Documentation 28(1): 11-21.

P.D. Turney, P. Pantel. 2010. From frequency to meaning: vector space models of semantics. Journal of Artificial Intelligence Research 37(1):141-188

Enhancing Multimodal Embeddings with Word Semantic Relations for Image Search Applications[*][†]

Mejorando representaciones de baja dimensionalidad con relaciones semánticas de palabras para aplicaciones de búsqueda de imágenes

Marco A. Gutiérrez
Robolab, University of Extremadura
Avda. de la Universidad, Cáceres, Spain
marcog@unex.es

Resumen: La generación de leyendas para imágenes juega un papel esencial en las aplicaciones de búsqueda de imágenes ya que nos permiten generar automáticamente descripciones de imágenes. Sin embargo a veces las palabras en estas leyendas generadas no son exactas y además pueden encontrarse abiertas a criticas subjetivas. También cuando buscan una imagen, los usuarios puede que no usen exactamente las mismas palabras que las existentes en esas leyendas sino otras con cierta similitud semántica. Por lo tanto presentamos un trabajo en el que expandimos el ámbito de nuestras leyendas generadas a partir de imágenes comparando la relación semántica entre la consulta y las palabras en la leyenda. En este trabajo usamos un pipleine codificador-decodificador que unifica representaciones de baja dimensionalidad de modelos imagen-texto con modelos de lenguage multimodales neuronales para generar descripciones de imágenes. Luego extendemos la semántica de estas descripciones utilizando vectores de palabras entrenados sobre grandes conjuntos de palabras para representar eficientemente su similitud semántica. Finalmente mostramos que haciendo uso de estas relaciones semánticas entre palabras somos capaces de encontrar conceptos mostrados en las imágenes que no estaban directamente escritos en las descripciones generadas incialmente.
Palabras clave: Representaciones de baja dimensionalidad, relaciones semánticas, Redes neuronales convolutivas, Vectores de palabras, Búsqueda de imágenes

Abstract: Image caption generation play a key role in image search applications as they allow us to automatically generate language based description of pictures. However sometimes the words on these generated captions might not be accurate and the result is open to criticism of subjectivity. Also, when searching for an image, users might not use the exact same words as the ones in generated captions but others with a semantic similarity. Therefore we present a work were we expand the scope of our image generated captions by looking at the semantic relation between the query and the words in the captions. We use an encoder-decoder pipeline that unifies joint image-text embedding models with multimodal neural language models to generate image captions. Then we extend the semantics of those captions making use of word vectors trained over large word datasets in order to effectively represent word semantic similarity. We finally show that by making use of these word semantic relations we are able to find concepts shown in the image that were not directly written in the initially generated captions.
Keywords: Embedding, Semantics, Convolutional Neural Networks, Word Vectors, Image search

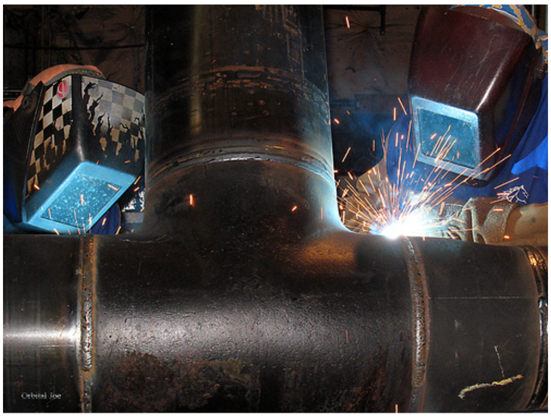
1 Introduction

Words can have multiple degrees of similarity (Mikolov, Yih, and Zweig, 2013). On top of that different users might query in different ways when looking for the same thing. Also systems might label images and generate descriptions in different manners that can even

[*] Thanks to Dr. Rafael E. Banchs from the Institute for Infocomm Research for his advice and support on this work.

[†] The author conducted this work as part of his A*STAR Research Attachment Programme (ARAP) at the Human Language Technology Department of Institute for Infocomm Research, Singapore.

be subjectively considered proper or not. Expanding the semantic scope of these image descriptions and the users queries can benefit the search of images producing a wider and more accurate range of results.



there is a painting of a pipe next to a glass wall.
a fire hydrant that is painted on top of a white wall.
a fire hydrant has painted green and white.
a fire hydrant has painted green and white.
a fire hydrant sitting next to a wall.

Figure 1: Top result of our system when looking for a picture with *smoke*. Note that the word *smoke* does not appear in the generated captions but there is still smoke on the picture.

Recent works like (He et al., 2015), (Vinyals et al., 2014) or (Xu et al., 2015) prove the big advances that have been done automatically generating captions for images. However these captions are usually short and, even though they could provide accurate descriptions, they do not contain all the information that is showed in the picture. Same objects can be described with different words. Therefore people can differ on how they would call something in an image. In order to extend the information contained on these generated sentences semantic relations between words can be exploited.

There are several techniques that provide semantic similarity between words (Christoph, 2016). Some approaches exploit manually created ontologies or taxonomies like WordNet (Fellbaum, 1998) or Freebase (Bollacker et al., 2008). These ontologies are manually created and maintained, sometimes being very costly. In consequence, only a few domains have a suitable ontology, limiting the applicability of similarity measures based on one of them. Dense vector representation approaches exploit the statistics over large text corpora by representing words as high dimensional sparse word count vectors. We use the skip-gram negative sampling approach (Mikolov et al., 2013b). These models are trained using windows extracted from a natural language corpus (i.e. an unordered set of words which occur nearby in a text sequence in the corpus). This allows us to easily retrain the system with new word scopes to cover new semantic areas. The final model is trained to predict, given a single word from the vocabulary, those words that will likely occur nearby in a text.

The system presented in this work weights the semantic relations between a query and image generated captions in order to improve the ranking of images to produce a result on a possible image search application. Therefore when a query is submitted to the system, nouns and adjectives from the query and from the captions are selected using the Natural Language Toolkit (NLTK) (Bird, Klein, and Loper, 2009). The neural network encoder-decoder pipeline described in (Kiros, Salakhutdinov, and Zemel, 2014) generates captions that describe a set of images. Then pre-trained word vectors helps finding semantic similarities between words on the captions and the ones selected form the query using the Skip-gram model described in (Mikolov et al., 2013a). Those similarities are calculated using the cosine distance in the vector space between the selected words in the query and the ones in the captions. Results are sorted by their calculated similarity weight, the best ones would be the ones with the highest similarity value. This process allows the expansion of the semantic domain of the words on the image generated captions being able to find things that are not explicitly noted in those sentences. Even in the case of querying for something that is not on the image dataset, the output will be more relevant than a random ordering of the images.

2 System design

Given a query our system otputs the top images that are most likely to contain what is described in the query. It accepts queries in the form of “get me a cup” or longer ones like “look for a cup on a table”. As shown in Figure 2, the system contains two main embedding subsystems. A multimodal encoder-decoder pipeline that generates the captions for a set of images and a word vector representation for the word semantic expansion. Words from the captions and the query are weighted on their semantic similarity and images are sorted on the average semantic value.

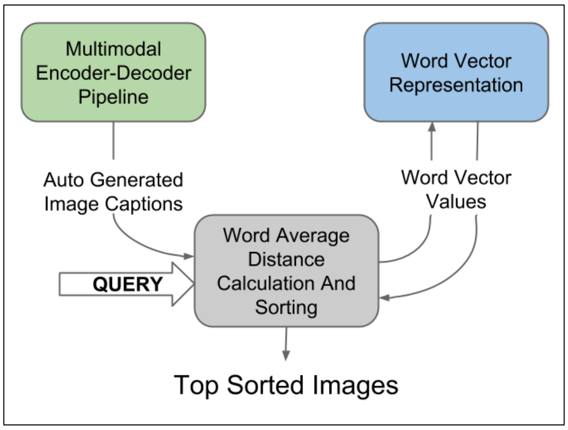

Multimodal Encoder-Decoder Pipeline
Word Vector Representation
Auto Generated Image Captions
Word Vector Values
QUERY
Word Average Distance Calculation And Sorting
Top Sorted Images

Figure 2: System architecture.

2.1 Multimodal encoder-decoder pipeline

This system is able to generate realistic image captions. The encoder is learned with a joint image-sentence embedding where sentences are encoded using long short-term memory (LSTM) recurrent neural networks (Hochreiter and Schmidhuber, 1997) Image features from the top layer of a deep convolutional network trained from the ImageNet classification task (Krizhevsky, Sutskever, and Hinton, 2012) are projected into the embedding space for the LSTM hidden states. A pairwise ranking loss is minimized in order to learn to rank images and their descriptions. For decoding the structure-content neural language model (SC-NLM) described in (Kiros, Salakhutdinov, and Zemel, 2014) is used which takes into account the content in the sentences.

2.2 Word Semantics Relationships

We decided to use neural networks for this task as they perform better than Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997) for preserving linear irregularities among words and in terms of computational cost when dealing with large training datasets (Mikolov, Yih, and Zweig, 2013) (Zhila et al., 2013). We use an improved version of the Skip-gram model (Mikolov et al., 2013a) to find word representations that predict the surrounding words in a document. The version used here makes use of negative sampling (Mikolov et al., 2013b) instead of the hierarchical softmax which tries to differentiate data from noise by means of logistic regression. With this, we build a word vector space that encodes semantics relations on the words of the training data. This semantic relationships are used in our system to weight the semantic relation through the cosine distance of these words.

2.3 Word matching system

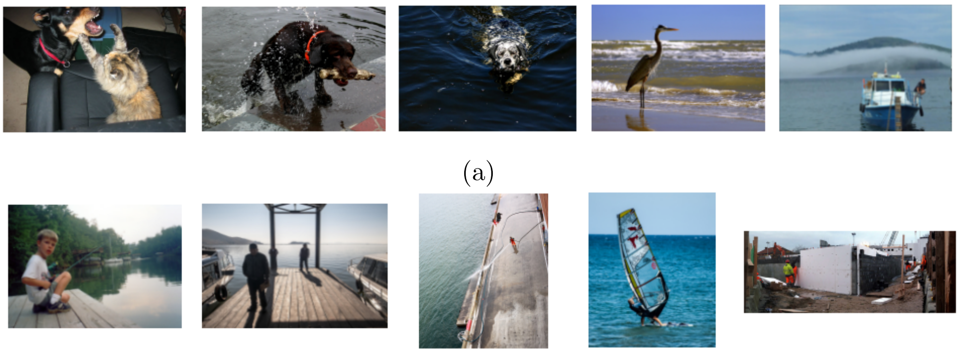
As a query comes it gets analyzed using NLTK and the nouns and adjectives are extracted. This words are the ones that will be used, since we consider them the most relevant on the query. The semantic weight of an image k is obtained by calculating the average of the cosine distance in vector space from each name or adjective from the query to each name or adjective in the top 5 generated captions of that image. Equation 1 shows the formal expression of this calculation, where n is the number of nouns in the query, m the number of nouns in the captions and d_{ij} is the cosine distance from word i from the query to word j on the captions.

$$W_k = \frac{1}{n+m}\sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij} = \frac{1}{n+m}\left(d_{11} + \cdots + d_{nm}\right) \tag{1}$$

Finally when all images weights are computed for the given query they are ranked by their weight value. The ones with the highest score will be the images whose captions have a highest semantic similarity to the query.

3 Experiments

As stated by (Besser, 1990) among others, a manual interpretation of the contents of an image will always be open to criticism of subjectivity. Therefore the difficulty of quantitatively evaluate the retrieval effectiveness of our approach. We tested our system against a direct-match approach where instead of using our semantic matching system the words are just directly matched. In this approach for each of the nouns and adjectives from the query that appear on the generated captions of an image will add a value of 1 to the weight of that image otherwise 0 will be added. This will end up with



(a)

(b)

Figure 3: Top results of the query *look for a pet in a river*, being the first the top left one and last the down right one. a) These are the results using the word semantics relations. None of the generated captions specifically contained ether the word *pet* or *river*. b) These are the results for the direct matching experiment. Only five of all the captions contained the word river. The rest are not shown since they all got a score of 0.

an image-to-query similarity weight equal to the number of nouns and adjectives they share. As on our system, the images are sorted by weight and those with a higher weight will be the top result of the approach. We show for each query the top results and the generated captions of our approach versus those with the direct matching approach. Due to space limits we can only show some results, for a wider overview please refer to: http://magutierrez.com/semantics-embeddings

For the experiments the LSTM encoder and SC-NLM decoder of the pipeline described in Section 2.1 have been trained on a concatenation of training sentences from both Flickr30K (Plummer et al., 2015) and Microsoft COCO (Lin et al., 2014). A subset of 1000 images from Flickr30K set is randomly selected and used for caption generation. These are the ones that will be used as possible results for the final answer to the query. Word representations in vector space are trained on part of Google News dataset (about 100 billion words). The final model contains 300-dimensional vectors for 3 million words and phrases.

Figure 1 shows the top result of searching for the word *smoke*. Not any of the generated captions show the word *smoke* among their results. Actually none of the captions contain the word smoke so the result of the direct-match approach is just a random ordering of the images with no sense at all. However our semantic based matching approach is able to detect the high similarity between fire and smoke and rank most of the pictures with fire on it with a higher similarity value. This way we can infer from the captions things that have a high probability of being in the picture even though they are not directly written there.

Figure 3 shows the results for the query *look for a pet in a river*. This query is longer and contains more words to evaluate. As a result we can see the direct match could find some *river* matches but probably not that much for pet. However our algorithm was able to evaluate the semantic relation between the word *pet* and some animals.

4 Conclusions and Future Work

Our system generates captions from images and expands their semantic scope using word representations in vector spaces. We have shown that weighting the words semantics relation of a query to the captions can significantly improve the results of an image search application. The system can even provide meaningful results when queried with words that don't even appear on the captions. Still different types of distances and weighting can be tested and compared in order to try to improve the results of the final image ranking. Further analysis ether of the query or the captions can be done with different natural language processing tools to determine the importance of the words and weight accordingly. Finally different ways of semantical relation among words can be also explored to extend and compare the results among the different relating approaches.

References

Besser, Howard. 1990. Visual access to visual images: the uc berkeley image database project. *Library Trends*, 38(4):787–798.

Bird, Steven, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python.* " O'Reilly Media, Inc.".

Bollacker, Kurt, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *In SIGMOD Conference*, pages 1247–1250.

Christoph, LOFI. 2016. Measuring semantic similarity and relatedness with distributional and knowledge-based approaches.

Fellbaum, Christiane. 1998. *WordNet.* Wiley Online Library.

He, Xiaodong, Rupesh Srivastava, Jianfeng Gao, and Li Deng. 2015. Joint learning of distributed representations for images and texts. *CoRR*, abs/1504.03083.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kiros, Ryan, Ruslan Salakhutdinov, and Richard S. Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Landauer, Thomas K and Susan T Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*. Springer, pages 740–755.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Plummer, Bryan, Liwei Wang, Chris Cervantes, Juan Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *CoRR*, abs/1505.04870.

Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044.

Zhila, Alisa, Wen-tau Yih, Christopher Meek, Geoffrey Zweig, and Tomas Mikolov. 2013. Combining heterogeneous models for measuring relational similarity. In *HLT-NAACL*, pages 1000–1009.

Towards phrase table expansion using automatically induced bilingual lexica

Jingyi Han
Universitat Pompeu Fabra
Roc Boronat, 138
08018 Barcelona
jingyi.han@upf.edu

Núria Bel
Universitat Pompeu Fabra
Roc Boronat, 138
08018 Barcelona
nuria.bel@upf.edu

Abstract

Bilingual lexica are crucial components of machine translation systems. In this paper, we propose a novel method to generate bilingual dictionaries by training a supervised classifier and to use the results to expand the phrase table of a statistical machine translation system. On average, we obtained precision, recall and f-score results above 0.92 for two language pairs, English-Spanish and Chinese-Spanish. The resulting lexica can then be directly used for phrase table expansion.

1. Introduction

Bilingual lexica are the key resource of statistical machine translation systems. Bilingual lexicon induction is the task of automatically providing such a resource from monolingual texts. Most of the research is based on the availability of large parallel or comparable corpora. However, such large bilingual related corpora are not readily available for many language pairs. In this article, we show how we produced bilingual lexicons out of large monolingual, and not related, corpora, to expand the phrase table of a statistical machine translation for the language pairs Chinese-Spanish and English -Spanish.

Recently, Mikolov et al. (2013b) showed that Word Embedding methods indeed project word semantics into a vector space from their distributional characteristics. More interestingly, it is claimed that the relationship between vector spaces that represent different language word semantics can be captured by a linear transformation. Therefore, we used word embedding vectors of translation equivalent word pairs to train classifier which can predict whether a new pair of words is under a translation relation or not.

The classifier learning curve shows that with a rather small quantity of positive examples (about 300 and 5:1 negative random examples) it is possible to achieve 90% accuracy for two quite different language pairs: English-Spanish and Chinese-Spanish. The results are especially encouraging because for the Chinese-Spanish language pair there are not many parallel or comparable corpora to exploit.

The rest of the paper is structured as follows: section 2 reports the previous works related to our approach; section 3 describes our supervised bilingual lexicon induction method; section 4 explains how the supervised learning technique is applied for expanding phrase table of SMT; section 5 sets the experimental framework; and finally, in section 6 results and conclusions are presented.

2. State of the art

Different previous works have shown how to learn bilingual lexica from non-parallel corpora, but still comparable corpora, i.e. collections of source-target document pairs that are not direct translations but are topically related. For instance, Yu and Tsujii (2009) extracted bilingual lexica from comparable corpora by considering the similarity of syntactic dependencies. Matsumoto et al. (2013) generated dictionaries by combining topic modeling and alignment techniques. Ananiadou et al. (2014) extracted bilingual terminology from comparable corpora using compositional and contextual clues. The main limitation of these approaches is that they are not usable when no large comparable corpora are available.

Recently, several interesting works treat bilingual lexicon generation as a supervised classification problem. For example, Irvine and Callison-Burch (2013) employed a supervised approach (linear classifier trained by stochastic gradient descent to minimize squared error) and combined extra-linguistic monolingually-derived signals (contextual, temporal, topical, orthographic, and frequency) as features for the model. Training data consist of 1250 positive examples and 3 times as many negative examples. The results are delivered in the form of ranked lists of English translations for 22 languages achieving very different top-10 accuracy rates: best results are for Spanish with 85% and the worst for Nepali with 13.6%. Differences are not related, though, to the amount of monolingual data available and the learning curve shows that performance is stable after about 300 positive training instances. The approach presented here is similar to Irvine and Callison-Burch (2013), but uses only linguistic features to train an SVM classifier. Our classifier achieves better results both in precision and recall and, interestingly, needs no extra-linguistic data. Our method basically trains a translation model using as features the word embedding distributed representations as proposed by Mikolov et al. (2013).

Word embedding vector representation has been shown to afford relevant distributional information in different semantic tasks: word similarity judgments and word analogy detection (Baroni et al., 2014; Levy et al., 2015, among others). Mikolov et al. (2013) in particular proposed using this distributed representation to automate the process of generating dictionaries and phrase tables for SMT. Their method learns a linear projection between vector spaces of two particular languages, a translation matrix, on the data provided by a 5K seed dictionary. At test time, a new word can be translated by projecting its vector representation from the source language space to the target language space. Once the vector in the target language space is obtained, similar target language vectors (found by cosine similarity assessment) are ranked as possible translations. The translation matrix is found via optimization with a stochastic gradient descent algorithm. Their results in the form of ranked lists are further refined with a confidence threshold that tries to balance precision and recall, i.e. coverage. Thus, the highest coverage achieved for the pair English-Spanish is 92.5%, but precision at top position is 53%. Best precision reported is 78% (better results are obtained when refining with edit distance) but with a coverage of 17%.

3. Supervised bilingual lexicon generation with word embeddings

Our supervised scenario is similar to SMT phrase table generation, where pairs of words are extracted from all possible combinations of words occurring in a given set of sentences and the probability of a particular word being the translation of another is estimated. With our method, all the words from two monolingual corpora can be proposed as translation candidates. Then, the classifier selects those pairs that are indeed possible translation equivalents, discarding many others. Note that in our approach only a training dataset made of word embeddings of actual translation equivalents is required.

Each word pair is represented by concatenating the word embedding vector representation of the source word and of its corresponding translation equivalent, for positive examples, and of random words for negative ones. Given a translation word pair (x, y), whose member vector features are $v(x) = (x_1, x_2, \ldots, x_n)$ and $v(y) = (y_1, y_2, \ldots, y_n)$ respectively, then $v(x, y)$ is defined as the concatenation of $v(x)$ and $v(y)$: $v(x, y) = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)$.

4. Phrase table expansion

The performance of SMT is always affected by parallel data shortage problem. Since our classifier can predict translation pairs from non-related monolingual corpora, it can efficiently alleviate such problem; the new translation pairs found by the classifier can be perfectly adapted as a Moses phrase-table, by using the confidence scores generated from the classifier for each word pair as translation model probabilities.

5. Methodology

In this section, we describe the experimental settings. After training a Sequential minimal optimization (SMO) based classifier using WE word pair vectors, we used the new translation pairs generated by the supervised classifier to expand the phrase table of a previously trained SMT system.

5.1. Data sets

5.1.1. Supervised classifier training.

We conducted our experiments on Chinese - Spanish (CH-ES) and English - Spanish (EN-ES) pairs. The monolingual corpora used were: Chinese Wikipedia Dump corpus [1] (54M words), for English, the BNC (100M) and a Spanish Wikipedia corpus [2] (120M, 2006 dump). Despite the fact that Spanish and Chinese corpora are Wikipedia dumps, there is no intended topic overlap. Also note that Chinese corpus is much smaller than the Spanish one; therefore, they cannot be considered neither parallel nor comparable.

To obtain a *translation* list (or positive class) for training and testing, we first randomly extracted about 500 words for each different PoS, noun, verb and adjective, from the ES monolingual corpus. These randomly selected words were then translated to target language words (EN and CH) using on-line Google Translate (following Mikolov et al., 2013, settings). Since not all the translations produced could be found in the target monolingual corpus, we removed from our datasets those words whose corresponding translation was not in the target corpus because we needed to obtain a word embedding. To build the *no translation* list, we randomly selected non-related source and target nouns from the monolingual corpus of each language and casually combined them. The ratio was 5 negative instances for each positive example. Final figures of the datasets are provided in table 1.

	CH-ES				EN-ES			
	Training		Testing		Training		Testing	
	YES	NO	YES	NO	YES	NO	YES	NO
Noun	449	2379	94	469	449	2379	94	469
Adj.	300	1500	99	500	300	1500	99	500
Verb	300	1500	99	500	300	1500	99	500
Total	1049	5379	292	1469	1049	5379	292	1469

Table 1: Translation pair datasets for CH-ES and EN-ES

5.1.2. Phrase table expansion of Phrase-based SMT

We trained two baselines on EN-ES and CH-ES using Moses (Koehn et al., 20). The parallel corpora used were: ES-EN News-Commentary corpora[4] (9.2M words) and CH-ES OpenSubtitles2013 parallel corpora[5] (9.5M words) for training; ES-EN News-test2011 corpora[6] (154K words) and CH-ES Open Subtitles 2012 parallel corpora[7] (7K words) for testing.

5.2 Word Embeddings

We obtained word embeddings for the Spanish, English and Chinese words in the translation and no translation lists using the Continuous Bag-of-words (CBOW) method as implemented in word2vec tool[3]. CBOW was chosen because it is faster and more suitable for larger datasets (Mikolov et al., 2013a), so in this work, we use CBOW model to learn the models of monolingual corpora. To train the CBOW models we used the default parameters with window size 8, minimum word frequency 5 and 200 dimensions for all vectors. To PoS tag the different corpora, we used Stanford PoS Tagger (Toutanova et al., 2003).

5.3 Classifier

We used the sequential minimal optimization algorithm (SMO, Platt, 1998) as implemented in Weka (Hall et al., 2009) for training a sup-

[1] https://archive.org/details/zhwiki_20100610
[2] http://hdl.handle.net/10230/20047

[3] http://code.google.com/p/word2vec/
[4, 6]http://www.statmt.org/wmt13/translation-task.html#download
[5, 7]http://opus.lingfil.uu.se/

port vector classifier. For evaluation, we divided the datasets into training and a hold-out test set.

5.4 New translation rules generation

To augment the original phrase table, the confidence score for each word pair was extracted from the classifier and was applied as its corresponding translation model probability. However, the existing phrase table has four features for all the bilingual phrase pairs. Since only one score can be obtained to weight our new translations, we used it as lexical weighting for the source and target words of both forward and backward directions. We assumed all our new pairs only occur once in the parallel corpus, so the translation model weights were scored with 1.

6 Results and conclusion

6.1 Supervised classifier prediction

We built and tested our SMO classifier for EN-ES and CH-ES for three word categories: noun (N), adjective (Adj) and verb (V) and another for the three categories together. The evaluation was double, as we performed a 10 fold cross-validation with the training test set and we tested again the model with the hold-out test set. Table 3 and 4 show the classification results in terms of precision (P), recall (R) and F1-measure (F).

CH-ES	**10 cross-validation**			**Hold-out test set**		
	P	**R**	**F**	**P**	**R**	**F**
N	0.947	0.948	0.948	0.933	0.934	0.931
Adj	0.916	0.918	0.917	0.934	0.936	0.932
V	0.955	0.956	0.955	0.957	0.958	0.958
All	0.927	0.928	0.927	0.941	0.942	0.941

Table 3: Results for CH-ES

EN-ES	**10 cross-validation**			**Hold-out test set**		
	P	**R**	**F**	**P**	**R**	**F**
N	0.963	0.963	0.963	0.944	0.945	0.944
Adj	0.964	0.964	0.964	0.953	0.952	0.952
V	0.965	0.966	0.965	0.927	0.93	0.928
All	0.922	0.924	0.922	0.921	0.922	0.921

Table 4: Results for EN-ES

Both pairs of languages show similar classification results superior to 0.9 F1 for all word categories obtained separately or all together, also with the hold-out test set. Results in Table 3 and 4 when further inspected showed that the performance of the classifier for the "right translation class" were worse than for the "no translation class". F1 scores for right translation class in case of *All* experiment were: 0.819 for CH-ES and 0.758 for EN-ES, most probably due to the smaller training set.

6.2 Phrase table expansion

We used the correctly classified right translation pairs to expand the phrase table: 215 translation pairs for EN-ES, and 229 translation pairs for CH-ES. In order to test the impact of the use of the classifier confidence as translation model probabilities, we first removed all the translation rules that contain our new word pairs from the original phrase table and we tested the new phrase table using BLEU metric (Papineni et al., 2002) with the same translation test set to see the differences. The results are shown in Table 5 and 6:

BLEU	Ave.	unigram	bigram	trigram
Moses PhT	24.17	59.9	30.0	17.6
Expanded PhT	24.43	60.2	30.2	17.8

Table 5: BLEU test results for the language pairs ES-EN

BLEU	Ave.	Ungram	bgram	trgram
Moses PhT	19.3	49.0	24.7	14.1
Expanded PhT	19.48	49.4	24.9	14.2

Table 6: BLEU test results for the language pair CH-ES

Since we only augmented with around 200 translation pairs into the phrase table, the improvement of the results is not obvious. But the goal of the present experiment was to show that the new translation rules generated with our supervised classifier can be used to expand the phrase table. Results showed that indeed it is a possible approach.

Acknowledgements

This work was funded with the support of the SUR of the DEC of the Generalitat de Catalunya and the European Social Fund, and by the project SKATER TIN2012-38584-C06-05.

References

Ahmet Aker, Monica Paramita and Rob Gaizauskas. 2013. Extracting bilingual terminologies from comparable corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics.*

Marco Baroni, Georgiana Dinu, and German Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the Conference of the Association for Computational Linguistics*.

Ann Irvine and Chris Callison-Burch. 2013. Supervised Bilingual Lexicon Induction with Multiple Monolingual Signals. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics*.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation, Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic, June 2007.

Georgios Kontonatsios, Ioannis Korkontzelos, Jun'ichi Tsujii and Sophia Ananiadou. 2014. Combining String and Context Similarity for Bilingual Term Alignment from Comparable Corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

Omer Levy and Yoav Goldberg. 2014b. Linguistic regularities in sparse and explicit word representations. In *Proceedings of CoNLL*.

Xiaodong Liu, Kevin Duh and Yuji Matsumoto. 2013. Topic Models + Word Alignment = A Flexible Framework for Extracting Bilingual Dictionary from Comparable Corpus. *In Proceedings of the Seventeenth Conference on Computational Natural Language Learning.*

Tomas Mikolov, Quoc V. Le and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. In *Proceedings HLT-NAACL*.

Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. 2013b. Efficient estimation of word representations in vector space. In *arXiv* preprint arXiv:1301.3781, 2013.

John C. Platt. 1998. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. *Advances in Kernel Methods: Support Vector Learning.*

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics*.

Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation (PDF). ACL-2002: 40th Annual meeting of the Association for Computational Linguistics. pp. 311–318.

Reinhard Rapp. 1995. Identifying word translations in nonparallel texts. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics.*

Ralf Steinberger, Bruno Pouliquen and Johan Hagman. 2002. Cross-lingual Document Similarity Calculation using the Multilingual Thesaurus EUROVOC. In *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics.*

Kristina Toutanova, Dan Klein, Chris Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics.*

Kun Yu and Junichi Tsujii. 2009. Extracting Bilingual Dictionary from Comparable Corpora with Dependency Heterogeneity. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics*.