# Towards a Vecsigrafo: Portable Semantics in Knowledge-based Text Analytics

Ronald Denaux[1] and Jose Manuel Gómez-Pérez[1]

Expert System, Madrid, Spain,
{rdenaux,jmgomez}@expertsystem.com

**Abstract.** The proliferation of knowledge graphs and recent advances in Artificial Intelligence have raised great expectations related to the combination of symbolic and distributional semantics in cognitive tasks. This is particularly the case of knowledge-based approaches to natural language processing as near-human symbolic understanding and explanation rely on expressive structured knowledge representations that tend to be labor-intensive, brittle and biased. This paper reports research addressing such limitations by capturing as embeddings the semantics of both words and concepts in large document corpora. We show how the emerging knowledge representation – our Vecsigrafo – can drive semantic portability capabilities that are not easily achieved by either word embeddings or knowledge graphs on their own, supporting curation, overcoming modeling gaps, enabling interlinking and multilingualism. In doing so, we also share our experiences and lessons learned and propose new methods that provide insight on the quality of such embeddings.

## 1 Introduction

For several decades, semantic systems have been predominantly developed around knowledge graphs (KGs) (and their variants: semantic networks and ontologies) at different degrees of expressivity. Language technologies, particularly knowledge-based text analytics have heavily relied on such structured knowledge. Through the explicit representation of knowledge in well-formed, logically sound ways, KGs provide rich, expressive and actionable descriptions of the domain of interest through logical deduction and inference, and support logical explanations of reasoning outcomes. On the downside, KGs can be costly to produce leading to scalability issues, as they require a considerable amount of well-trained human labor [9] to manually encode knowledge in the required formats. Capturing the knowledge from the crowd has been suggested[1], but scalability is proportional to the number of humans contributing to such tasks. Furthermore, the involvement of humans in the modelling activities introduces a bias in how the domain is represented (its depth, breadth and focus) which can lead to brittle systems that only work in a limited context, hinders generality and may require continuous supervision and curation.

In parallel, the last decade has witnessed a shift towards statistical methods due to the increasing availability of raw data and cheap computing power.

Statistical approaches to text understanding have proved to be powerful and convenient in many linguistic tasks, such as part-of-speech tagging, dependency parsing, and others. However, these methods are also limited and cannot be considered as a replacement for knowledge-based text analytics. E.g. humans seek causal explanation which are hard to provide based on statistical methods, as they are driven by statistical induction rather than logical deduction.

Recent results in the field of distributional semantics [13] have shown promising ways to learn features from text that can complement the knowledge already captured explicitly in KGs. Embeddings provide a compact and portable representation of words and their associated semantics that stems directly from a document corpus. Here, the notion of *semantic portability* refers to the capability to capture as an information artifact (a vector) the semantics of a word from its occurrences in the corpus and how such artifact enables that meaning to be merged with other forms of (possibly structured) knowledge representation.

At Expert System, we make extensive use of formal knowledge representations, including semantic networks and linguistic rule bases, as these technologies have shown higher accuracy figures if properly fine tuned, are more resilient against scarce training data and tend to offer better inspection capabilities allowing us to debug and adapt when necessary. However, this comes at the cost of considerable human effort by linguists and knowledge engineers for various *KG curation tasks*: continuous bug fixing, keeping resources up to date (e.g. *Barack Obama is/was the president of the US*), and adding extensions for new domains or terms of interests *(Cybersecurity, Blockchain)* for each language supported. We argue that semantic portability is a key feature to facilitate KG-curation tasks, deal with modeling bias, and enable interlinking of KGs.

In this paper, we present research and experiences evaluating, adopting and adapting approaches for generating and applying word and concept embeddings. Among others, we argue that using embeddings in practice (other than as inputs for further deep learning systems) is not trivial as there is a lack of evaluation tools and best practices other than manual inspection, which we are aiming to minimize in the first place. Therefore, we describe some methods we have developed to check our intuitions about these systems and establish reproducible good practices. Our main contributions are: i) a novel method for the generation of semantically portable, joint word and concept embeddings and their applications in hybrid knowledge-based text analytics (Sect. 3), ii) inspection and evaluation methods that have proved useful for assessing the quality and fitness of embeddings for the purpose of our research (Sect. 4). This paper also applies the embedding generation to Expert System's case, resulting in a Vecsigrafo and describes a practical application for the Vecsigrafo (Sect. 5).

## 2 Background

NLP systems which perform *Knowledge-based Text Analysis* rely on a *Knowledge Graph* (KG) as its point of reference for performing analyses. Good KGs for text analysis represent concepts and entities, their semantic and grammatical relations, and their lexical forms enabling the system to recognise and disambiguate those concepts. KGs used in practice include DBpedia, Word- and BabelNet.

The Knowledge Graph is used by a *text analysis engine* which performs NLP tasks such as tokenization, part-of-speech tagging, etc. Furthermore, many text analysis engines can use the knowledge encoded in the KG to perform **word sense disambiguation** (WSD), in which particular senses are ascribed to words in the text. This can be done for simple cases, such as knowing whether *apple* refers to the company or the fruit; but this extends to more subtle differences such as distinguishing between *redeem* as "exchanging a voucher" or as "paying off a debt" or its religious sense. The sense disambiguation results can then be used to improve further NLP tasks such as *categorization* and *information extraction*; this can be done either using machine learning or rule-based approaches.

**At Expert System** our KG is called **Sensigrafo**[1] that relates concepts (internally called syncons) to each other via an extensible set of relations (e.g. hypernym, synonym, meronym), to their lemmas (base forms of verbs, nouns, etc.) and to a set of around 400 topic domains (e.g. *clothing*, *biology*, *law*, *electronics*). For historical, strategic and organizational reasons we maintain different Sensigrafos for the 14 languages we support natively; we have partial mappings at the conceptual level between some (but not all) of our Sensigrafos as producing and maintaining these mappings requires prohibitive amounts of human effort. The Sensigrafo is used by **Cogito**, our text analysis engine which performs word sense disambiguation with an accuracy close to 90% for languages with mature native support. On top of Cogito we have *rule languages* which allow us to write (or learn) custom *categorization and extraction rules* based on the linguistic characteristics (including disambiguated concepts) of documents.

## 2.1 Word and KG embeddings

Various approaches for statistical, corpus-based models of semantic representations have been proposed over the last two decades [3]. Traditionally encoded in sparse pointwise mutual information matrices, recent approaches generate *embeddings*: dense, low-dimensional spaces that capture similar information as the pointwise mutual information matrices. In particular, the `word2vec` system based on the skip-gram with negative-sampling (SGNS) algorithm [13] provides an efficient way to generate high-quality word embeddings from large corpora.

Although SGNS is defined in terms of sequences of words, subsequent algorithms such as GloVe [15] and Swivel [18] have shown that similar (or better) results can be achieved by learning the vectors from sparse co-occurrence matrices. These approaches make it easier to generalise over different types of linguistic units (e.g. concepts and words) as we show in Sect. 3, which is harder to do with SGNS since it expects non-overlapping sequences of linguistic units.

Standard corpus-based word embeddings do not encode KG-concepts due to ambiguity of words in natural language. One approach to resolve this is to generate *sense embeddings* [5,10], whereby tags are added to the words in the

[1] You can think of Sensigrafo as a highly curated version of WordNet

corpus to indicate the sense and part-of-speech of the word. While this addresses ambiguity of individual words, the resulting embeddings do not directly provide embeddings for KG-concepts, only to various synonymous word-sense pairs[2].

Approaches have been also proposed for learning concept embeddings from existing Knowledge Graphs [4,8,16]. Compared to corpus-based embeddings, We find that KG-derived embeddings are not yet as useful for our purposes because: (i) KG embeddings encode knowledge which is relatively sparse (compared to large text corpora); (ii) the original KG already is structured and is easy to query and inspect; (iii) corpus-based models provide a *bottom-up view* of the linguistic units and reflect how language is used in practice, as opposed to KGs, which provide a *top-down view* as they have usually been created by human experts (e.g. Sensigrafo, which has been hand-curated by linguists). Other proposed methods can generate *joint embeddings of words and KG entities* [20]. However, they mix bottom-up and top-down views[3] which we want to avoid. Hence, in Section 3, we propose a corpus-based, joint word-concept embedding generation method.

## 2.2 Existing Methods and Practices for Evaluating Embeddings

One evaluation method used in the literature and open-sourced tools relies on **manual inspection**: papers provide examples using a top-n of similar words for a given input word to show semantic clustering. Similarly, **visual inspection** is provided in the form of t-SNE or PCA dimensionality reduction projections into two dimensions which also show semantic clustering; an example of a tool that can be used for this purpose is the Embeddings Projector [19], distributed as part of Tensorflow. However, these tools restrict the view to the neighbourhood of a single point or area of the embedding space and make it hard to understand the overall behaviour of the embeddings. In Sections 4.2 and 4.3 we introduce a plot which addresses this issue and show concrete applications.

**Intrinsic evaluation** methods are used to try to understand the overall quality of embeddings. In the case of word embeddings, a few papers employ such methods to provide systematic evaluations of various models [2,12]. As part of these evaluations [2] defines 5 types of tasks (and lists available test-sets) that compare embedding predictions to human-rated datasets. From the five identified types, two (semantic relatedness and analogy) are consistently used in the recent literature, while the other three (synonym detection, concept categorization and selectional preference) are rarely used. Such intrinsic evaluations define specific, somewhat artificial tasks which are not end-goals of the embeddings. Typically, embeddings are generated to be used within larger (typically machine-learning based) NLP systems [17]; improvements on those NLP systems by using the embeddings provide *extrinsic* evaluations, which are rarely mentioned in the literature. Schnabel et al. [17] show that good results in intrinsic evaluations do not guarantee better extrinsic results. In Sections 4 and 5 we present both intrinsic and extrinsic evaluations.

[2] E.g. word-sense pairs $\texttt{apple}_2^{\texttt{N}}$ and $\texttt{Malus\_pumila}_1^{\texttt{N}}$ have separate embeddings, but the concept for *apple tree* they represent has no embedding.

[3] By aligning a TransE-like knowledge model and a SGNS-like text model

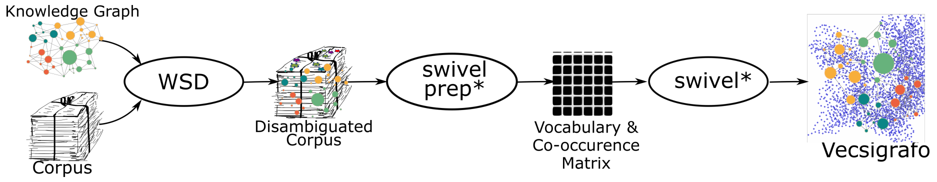


Fig. 1: Process for Vecsigrafo generation from a text corpus.

## 3 Corpus-based Joint Concept-Word Embeddings

In order to build hybrid systems which can use both bottom-up (corpus-based) embeddings and top-down (KG) knowledge, we propose to generate embeddings which share the same vocabulary as the Knowledge Graphs. This means generating embeddings for knowledge items represented in the KG such as concepts and surface forms (words and expressions) associated to the concepts in the KG[4].

The overall process for learning joint word and concept embeddings is depicted in Figure 1, we start with a text corpus on which we apply tokenization and word sense disambiguation (WSD) to generate a *disambiguated corpus*, which is a sequence of **lexical entries** (words, or multiword expressions). Some of the lexical entries are annotated with a particular sense (concept) formalised in the KG. To generate embeddings for both senses and lexical entries, we need to correctly handle lexical entries which are associated to a sense in the KG, hence we extend the matrix construction phase of the Swivel [18] algorithm to generate a co-occurrence matrix which includes both lexical forms and senses as part of the vocabulary as explained below. Then we apply the training phase of a slightly modified version of the Swivel algorithm to learn the embeddings for the vocabulary; the modification is the addition of a vector regularization term as suggested in [7] (equation 5) which aims to reduce the distance between the column and row (i.e. focus and context) vectors for all vocabulary elements.

**Modified Swivel Co-occurrence Matrix Construction** The main modification from standard Swivel[5] is that in our case, each token in the corpus is not a single word, but a lexical entry with an optional KG-concept annotation. Both lexical entries and KG-concepts need to be taken into account when calculating the co-occurrence matrix. Formally, the co-occurence matrix $X \in \mathbb{R}^{V*V}$ contains the co-occurrence counts found over a corpus, where $V \subset L \cup C$ is the vocabulary, which is a conjunction of lexical forms $L$ and KG-concepts $C$. $X_{ij} = \#(v_i, v_j)$ is the frequency of lexical entries or concepts $v_i$ and $v_j$ co-occurring within a certain window size $w$. Note that $X_{ij} \in \mathbb{R}$, since this enables us to use a dynamic context window [12], weighting the co-occurrence of tokens according to their distance within the sequences.[6].

[4] In RDF, this typically means values for `rdfs:label` properties, or words and expressions encoded as `ontolex:LexicalEntry` instances using the lexicon model for ontologies (see `https://www.w3.org/2016/05/ontolex/`).

[5] As implemented in `https://github.com/tensorflow/models/tree/master/swivel`

[6] We use a modified harmonic function $h(n) = 1/n$ for $n > 0$ and $h(0) = 1$ which covers the case where a token has both a lexical form and a concept. This is the

### 3.1 Vecsigrafo Generation at Expert System

Table 1: Size of vocabularies(×1000), for English and Spanish in Sensi- and Vecsigrafos

| Vocab Element | En-grafo | | Es-grafo | |
|---|---|---|---|---|
| | Sensi- | Vecsi- | Sensi- | Vecsi- |
| Lemmas | 398 | 80 | 268 | 91 |
| KG-Concepts | 300 | 67 | 226 | 52 |
| Total | 698 | 147 | 474 | 143 |

We follow the process described in Fig. 1 by adapting it to Expert System's technology stack (described in Sect. 2). As input we have used the UN corpus [21], specifically the alignment between English and Spanish corpora, which has about 22 million lines for each language. We used Cogito to perform tokenization, lemmatization and word-sense disambiguation. We use tokens at the disambiguation level, this means that some tokens correspond to single words, while others correspond to multi-word expressions (when they can be related to a Sensigrafo concept). Furthermore, we only kept lemmas (or base-form) as our lexical entries since the Sensigrafo only associates concepts to lemmas, not the various morphological forms in which they can appear in natural language; this reduces the size of the vocabulary. Also, we perform some filtering of the tokens by removing stopwords. We trained two vecsigrafos for Spanish and English for 80 epochs. The resulting vecsigrafos are summarised and compared to the corresponding Sensigrafos in Table 1. As the table shows, the UN corpus only covers between 20 and 34 % of the lemmas and concepts in the respective Sensigrafos.

## 4 Evaluating Vecsigrafos

In general, we find that generating embeddings is relatively easy by using and adapting existing tools. However, evaluating the quality of the resulting embeddings is not as straightforward. We have used both manual and visual inspection tools but ran into the issues discussed in Sect. 2.2. In particular, the Embeddings Projector [19] is limited to displaying 10K points, hence we can only visualize a part of the vecsigrafo at a time. By combining information from Sensigrafo and exploring areas of the space with this projector, we have been able to find some vocabulary elements which were "out of context"; this was typically caused either by a limitation of the corpus, or by issues with our language processing pipeline (e.g. tokenization or disambiguation), which could then be further investigated.

### 4.1 Semantic Relatedness

Testing on various semantic relatedness gives us results which are well below the state of the art as shown in Table 2. Part of this is due to the corpus used, which is smaller and more domain restricted than other corpora (e.g. Wikipedia,

same weighing function used in GloVe and Swivel; word2vec uses a slightly different function $d(n) = n/w$.

Table 2: Vecsigrafo Semantic Relatedness results compared to state of the art

| model | WSsim | WSrel | simlex999 | rarewords | simverb |
|---|---|---|---|---|---|
| SotA 2015 [12] | 79.4 | 70.6 | 43.3 | 50.8 | n/a |
| Swivel [18] | 74.8 | 61.6 | 40.3 | 48.3 | 62.8 |
| Swivel$_{\text{UNv1.en}}$ | 58.8 | 45.0 | 18.3 | 37.8 | 15.3 |
| Vecsigrafo$_{\text{UNv1.en}}$ | 47.6 | 24.1 | 12.4 | 30.8 | 13.2 |

Gigaword): when we apply standard Swivel on the UN corpus we see a substantial decrease in performance. Furthermore, the lemmatization and inclusion of concepts in the vocabulary may introduce noise and warp the vector space negatively impacting the results for this specific task.

Although these results are disappointing, we note that (i) the results only provide information about the quality of the lemma embeddings but not the concept embeddings; (ii) an easy way to improve these results is to train a Vecsigrafo on a larger corpus. Also, most of the available datasets are only available for English and we could not find similar datasets for Spanish.

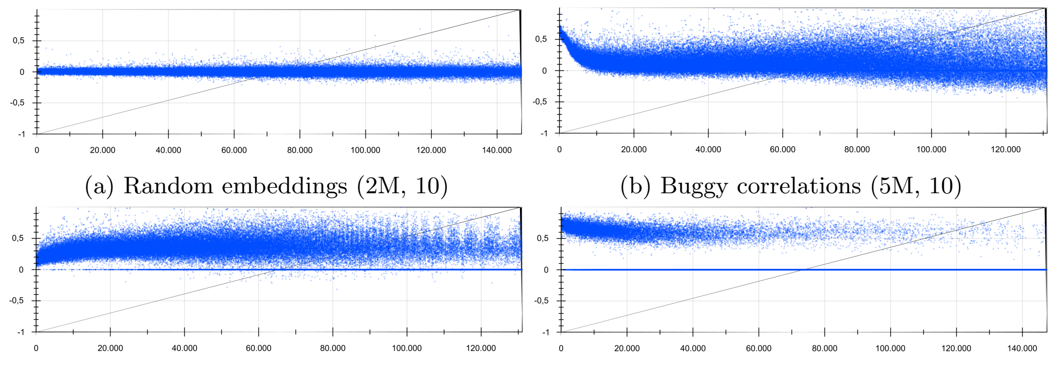
## 4.2 Word Prediction Plots

To address the limitations of top-n neighbour, visual exploration and semantic relatedness approaches discussed above, we developed a method to generate plots that can give us an overview of how well the embeddings perform across the entire vocabulary. We call these *word prediction plots*.

The idea is to simulate the original `word2vec` learning objective –namely predicting a focus word based on its context words (or vice versa)– while gathering information about elements in the vocabulary. To generate the plot, we need a *test corpus*, which we tokenize and disambiguate in the same manner as during vecsigrafo generation. Next, we iterate the disambiguated test corpus and for each token, we *calculate the cosine similarity* between the embedding of the focus token and the weighted average vector of the context vocabulary elements in the window. After iterating the test corpus we have, for each vocabulary element in the test corpus, a list of cosine similarity values for which we can derive statistics such as the average, standard deviation, minimum and maximum.

The plots are *configurable* as you can vary: (i) the statistic to display, e.g. average, minimum; (ii) how to display vocabulary elements(e.g. vary order or colour); (iii) versions of generated embeddings; (iv) test corpora (e.g. Wikipedia, Europarlament); (v) the window size and dynamic context window weighting scheme (see Sect. 3) (vi) the size of the test corpus: the larger the corpus the slower it is to generate the cosine similarities, but the more lexical entries will be encountered. Plot generation takes, on a typical PC, a couple of minutes for a corpus of 10K lines and up to half an hour for a corpus of 5M lines.

These types of plots are useful to verify the quality of embeddings and to explore hypotheses about the embedding space as we show in Figure 2. Fig. 2a shows a plot obtained by generating random embeddings for the English vocabulary; as can be expected the average cosine similarity is close to zero for the



(a) Random embeddings (2M, 10) (b) Buggy correlations (5M, 10)

(c) Uncentered (5M, 4) (d) Re-centered (10K, 5)

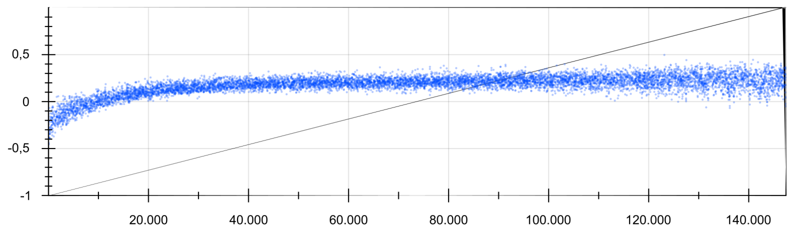
Fig. 2: Example word prediction plots, number of sequences of the test corpus used and the context window size. The horizontal axis shows the rank of the vocabulary elements sorted from most to least frequent; the vertical axis shows the *average cosine similarity* (which can range from -1 to 1).

150K lexical entries in the vocabulary. Fig. 2b shows a plot for early embeddings we generated where we had a bug calculating the correlation matrix. *Manual inspection of the embeddings seemed reasonable*, but the plot clearly shows that only the most frequent 5 to 10K vocabulary elements are consistently correct; for most of the remaining vocabulary elements the predictions are not better than random (although some of the predictions were good). Fig. 2c shows results for a recent version of the Spanish vecsigrafo, which are clearly better than random; although overall values are rather low. Fig. 2d shows the plot for our current embeddings where the vector space is re-centered as explained next.

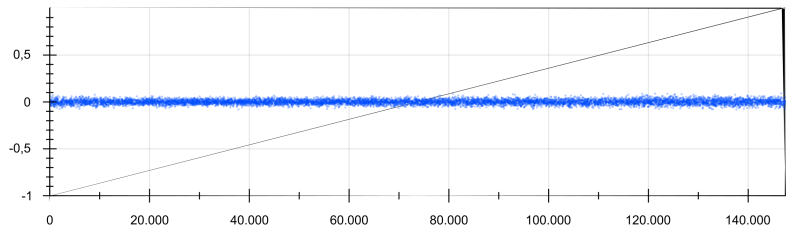
## 4.3 Vector Distribution and Calibration

One of the things we noticed using the prediction plots was that, even after fixing bugs with the co-occurrence matrix, there seemed to be a *bias against the most frequent vocabulary elements*, as shown in fig. 2c, where it seems harder to predict the most frequent words based on their contexts. We formulated various hypotheses to try to understand why this was happening, which we investigated by generating further plots. A useful plot in this case was generated by calculating the average cosine similarity between each vocabulary element and a thousand randomly generated contexts. If the vector space is well distributed, we expected to see a plot similar to fig. 2a. However, the result depicted in figure 3a *verifies the suspected bias* by showing that given a random context, the vector space is more likely to predict an infrequent word rather than a frequent one.

To avoid this bias, we can recalibrate the embedding space as follows: we calculate the centroid for all the vocabulary elements and then shift all the vectors so that the centroid becomes the origin of the space. When generating the random contexts plot again using this *re-centered* embedding space, we get the expected results. Figures 2c and 2d, show that this re-centering also improves the prediction of the most frequent lexical entries.



(a) Original (uncentered)



(b) Re-centered

Fig. 3: Average cosine similarity prediction plots for random contexts.

## 5 Vecsigrafo Application: Cross-KG Concept Alignment

As mentioned in Sect. 2, there are many tasks currently requiring manual effort, where a Vecsigrafo can be useful. In this section we discuss one such application: cross-KG alignment of concepts. Sensigrafos for different languages have been modelled by different teams and fit different strategic needs, hence they differ in terms of maturity and conceptual structure (besides the linguistic differences). This provides a use-case for semantic portability as, in order to support cross-linguality, we need to be able to map concepts between different Sensigrafos as accurately as possible. We describe how we apply vecsigrafos to accomplish this.

**Mapping Vector Spaces** We followed Mikolov et al.[14] approach to generate embeddings for different languages and then aligning the different spaces. To do this we need a *seed dictionary* between the vocabularies, which we had in the form of a partial mapping (for 20K concepts) between the Spanish and English sensigrafos. We expanded the partial concept mapping to generate a dictionary for lemmas (covering also around 20K lemmas). We split this dictionary into a training, validation and test set (80-10-10) and tried a couple of methods to derive an alignment between the spaces, summarised in Table 3. Although Mikolov suggests using a simple linear translation matrix (TM) [14], we found this method had very poor results. This suggests the desired alignment between the embedding spaces is highly non-linear[7], which prompted us to use neural

Table 3: Alignment method performance

| Method | Nodes | Hit@5 |
|---|---|---|
| TM | n/a | 0.36 |
| NN2 | 4K | 0.61 |
| NN2 | 5K | 0.68 |
| NN2 | 10K | 0.78 |
| NN3 | 5K | 0.72 |

[7] We explored and verified this in a number of experiments not reported in this paper.

networks (NN) with ReLU activations to capture these non-linearities. The best NN was able to include the correct translation for a given lexical entry in the top-5 of nearest neighbours in 78% of cases in our test set. In 90% of the cases we found that the top 5 suggested translations were indeed semantically close. The results indicate that the dictionary we used to train the alignment models covers the *low ambiguity* cases where straightforward ontology alignment methods can be applied.

Table 4: Manual inspection of bilingual embeddings

| | in dict | out dict |
|---|---|---|
| # concepts | 46 | 64 |
| hit@5 | 0.72 | 0.28 |
| no concept$_{\text{es}}$ | 2 | 33 |

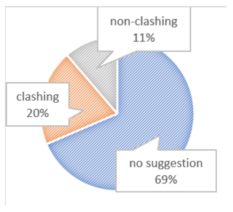
**Combining embedding and KG information** The results shown in Table 3 are very encouraging, hence we used them to generate a **bi-lingual vecsigrafo**. To check how well the bi-lingual vecsigrafo generalises from the dictionary, we took a sample of 100 English concepts and analysed their top5 neighbours in Spanish as shown in Table 4. The `hit@5` for the in-dictionary is in line with our test results; but for the out-of-dictionary concepts, we could only manually find an exact synonym in 28% of the cases. Manual inspection showed that for over half of the concepts, the corresponding Spanish concept had not been included in the vecsigrafo, or there was no exact match in the Spanish Sensigrafo. Furthermore, as Table 1 shows, the Spanish sensigrafo has 75K fewer concepts than English and due to modelling and language differences, many *concepts may be fundamentally unmappable*[8]. In conclusion: the bi-lingual vecsigrafo can help us find missing mappings, but still requires manual validation as it does not provide a solution to the underlying problem of *finding exactly synonymous concepts*.

Our next step was to design a **hybrid synonym concept suggester** which combines features from the bi-lingual vecsigrafo, the information in the Sensigrafos and PanLex [11], a multilingual thesaurus. In broad lines, the suggester works as follows: for a given concept in the source language, we find the $n$ nearest concepts in the target language *that match the grammar type* (i.e. they should be either nouns, verbs, adjectives, etc.); next, for each candidate, we calculate a set of hybrid features such as likelihood of lemma translation, glossa similarity, absolute and relative cosine similarity, shared hypernyms and domains. We then combine the various features into a single score, which we use to re-order the candidates and decide whether we have found a possible exact synonym. Finally, we verify whether the suggested synonym candidate is already mapped to a different concept and if so, we calculate the same features for this pair and compare it to the score for the top candidate.

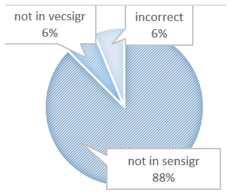
The output of the synonym suggester for an input is either (i) no suggestion or (ii) a suggested synonym, which can be either clashing or non-clashing. Figure 4a shows the output suggestions for 1546 English concepts, used in a large rulebase for IPTC categorization, for which we did not have a Spanish mapping. We manually inspected 30 of these cases to verify the suggestions. The results,

[8] The size and scope of the UN corpus limits the concepts available in Vecsigrafo.

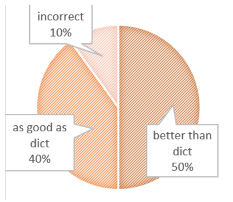
shown in Fig. 4, confirm that the suggestions are mostly accurate. In fact, for 5 of the clashing cases, the suggestion was *better* than the existing mapping; i.e. the existing mapping was not an exact synonym. For another 4 clashing suggestions, the existing mapping had very close meanings, indicating the concepts could be merged. In some cases suggestions were not exact synonyms, but pointed at modelling differences between the sensigrafos. For example, for the English concept `vote`, a verb with glossa *go to the polls*, the suggested Spanish synonym was concept `votar`, a verb with glossa *to express an opinion or preference, for example in an election or for a referendum*, which is more generic than the original concept. However, the Spanish Sensigrafo does not contain such a specific concept (among 5 verb concepts associated `votar`) and the English Sensigrafo does not contain an equivalent concept (among 6 verb concepts associated to `vote`, plus 9 non-verb concepts).



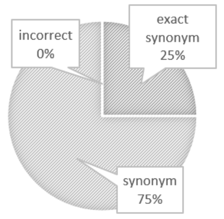
(a) 1546 suggestions



(b) 16 no suggestion



(c) 10 clashing



(d) 4 non-clashing

Fig. 4: Synonym suggestions for 1546 syncons and manual inspection breakdown.

## 6 Conclusion and Future Work

This paper introduced a method for generating joint word-concept embeddings based on word sense disambiguation which can be combined with knowledge graphs into a Vecsigrafo for providing hybrid capabilities that could not be achieved by either the KG or word-embeddings alone. We presented evaluation methods that we have used, introducing a new kind of plot for assessing embedding spaces. Finally, we presented a practical application of Vecsigrafo showing promising results. We believe these methods can be employed to improve KGs and tools in the Semantic Web and Computational Linguistics communities.

As future work, we intend to parallelise the presented pipelines for vecsigrafo and plot generation and apply these to larger corpora. We are also extending a rule translator [6] using Vecsigrafo to support new language combinations; previously, translations were only possible between fully-mapped, closely related languages (e.g. Italian to Spanish). We are designing Vecsigrafo-powered tools that can be used by our linguists to assist in Sensigrafo curation and alignment tasks.

**Acknowledgements** This work is supported by CDTI (Spain) as project IDI-20160805 and by the European Comission under grant 700367 – DANTE – H2020-FCT-2014-2015/H2020-FCT-2015.

## References

1. Aroyo, L., Welty, C.: Truth is a lie: Crowd truth and the seven myths of human annotation. AI Magazine 36(1), 15–24 (2015)
2. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In: ACL (2014)
3. Baroni, M., Lenci, A.: Distributional Memory: A General Framework for Corpus-Based Semantics. Computational Linguistics 36(4), 673–721 (2010)
4. Bordes, A., Usunier, N., Weston, J., Yakhnenko, O.: Translating Embeddings for Modeling Multi-Relational Data. Advances in NIPS 26, 2787–2795 (2013)
5. Chen, X., Liu, Z., Sun, M.: A Unified Model for Word Sense Representation and Disambiguation. In: EMNLP. pp. 1025–1035 (2014)
6. Denaux, R., Biosca, J., Gomez-Perez, J.M.: Framework for Supporting Multilingual Resource Development at Expert System. In: Meta-Forum. Lisbon (2016), `http://www.meta-net.eu/events/meta-forum-2016/slides/31_denaux.pdf`
7. Duong, L., Kanayama, H., Ma, T., Bird, S., Cohn, T.: Learning Crosslingual Word Embeddings without Bilingual Corpora. In: EMNLP-2016. pp. 1285–1295 (2016)
8. Feng, J., Huang, M., Yang, Y., Zhu, X.: GAKE: Graph Aware Knowledge Embedding. In: COLING. pp. 641–651 (2016)
9. Gunning, D., Chaudhri, V.K., Clark, P.E., Barker, K., Chaw, S.Y., Greaves, M., Grosof, B., Leung, A., McDonald, D.D., Mishra, S., Others: Project Halo Update—Progress Toward Digital Aristotle. AI Magazine 31(3), 33–58 (2010)
10. Iacobacci, I., Pilehvar, M.T., Navigli, R.: SENSEMBED: Learning Sense Embeddings for Word and Relational Similarity. In: 53rd ACL. pp. 95–105 (2015)
11. Kamholz, D., Pool, J., Colowick, S.M.: PanLex: Building a Resource for Panlingual Lexical Translation. In: LREC. pp. 3145–3150 (2014)
12. Levy, O., Goldberg, Y., Dagan, I.: Improving Distributional Similarity with Lessons Learned from Word Embeddings. Transactions of the ACL 3(0), 211–225 (2015)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: NIPS (2013)
14. Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting Similarities among Languages for Machine Translation. Tech. rep., Google Inc. (sep 2013)
15. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP. vol. 14, pp. 1532–1543 (2014)
16. Ristoski, P., Paulheim, H.: RDF2Vec: RDF graph embeddings for data mining. In: International Semantic Web Conference. vol. 9981 LNCS, pp. 498–514 (2016)
17. Schnabel, T., Labutov, I., Mimno, D., Joachims, T.: Evaluation methods for unsupervised word embeddings. In: EMNLP. pp. 298–307. ACL (2015)
18. Shazeer, N., Doherty, R., Evans, C., Waterson, C.: Swivel: Improving Embeddings by Noticing What's Missing. arXiv preprint (2016)
19. Smilkov, D., Brain, G., Thorat, N., Nicholson, C., Reif, E., Viégas, F.B., Wattenberg, M.: Embedding Projector: Interactive Visualization and Interpretation of Embeddings. In: Interpretable Machine Learning in Complex Systems (2016)
20. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge Graph and Text Jointly Embedding. EMNLP 14, 1591–1601 (2014)
21. Ziemski, M., Junczys-Dowmunt, M., Pouliquen, B.: The united nations parallel corpus v1. 0. In: Language Resource and Evaluation (2016)