Tackling Fairness, Change and Polysemy in Word Embeddings

Felipe Bravo-Marquez

Department of Computer Science, University of Chile
National Center for Artificial Intelligence Research
Millenium Institute Foundational Research on Data

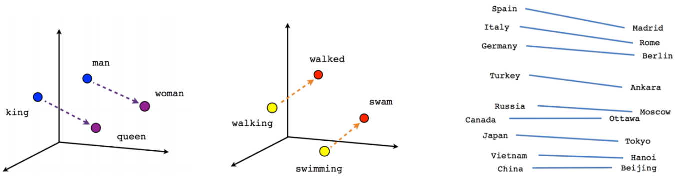
December 7, 2021

CIENCIAS DE LA COMPUTACIÓN UNIVERSIDAD DE CHILE
CENTRO NACIONAL DE INTELIGENCIA ARTIFICIAL
Millennium Institute Foundational Research on Data
RELELA Representations for Learning and Language

Word Embeddings

- The first step in computationally working with written language is to represent words as mathematical objects we can operate with.
- Representing words as numeric vectors a.k.a **embeddings** is a standard practice in **Natural Language Processing** (NLP).
- Word embeddings are a mapping of discrete symbols (i.e., words) to continuous vectors.
- Distance between vectors can be equated to distance between words.
- This makes easier to generalize the behavior from one word to another.
- Word embeddings have become a **core component** of NLP downstream systems (e.g., sentiment analysis, machine translation, question answering).

Word Embeddings



Male-Female Verb tense Country-Capital

Word embeddings can encode semantic and syntatic relationships between words.

Distributional Hypothesis

- The construction of word embeddings from document corpora is based on the **Distributional Hypothesis** [Harris, 1954]:

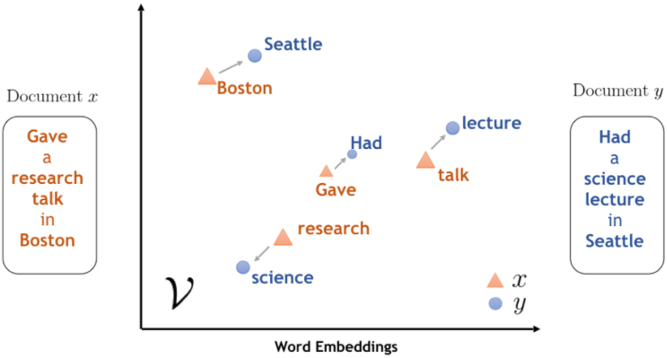
  > *Words occurring in the same **contexts** tend to have similar meanings.*

- Or equivalently:

  > *A word is characterized by the **company** it keeps.*

- The idea is to map words occuring in similar contexts to similar vectors.

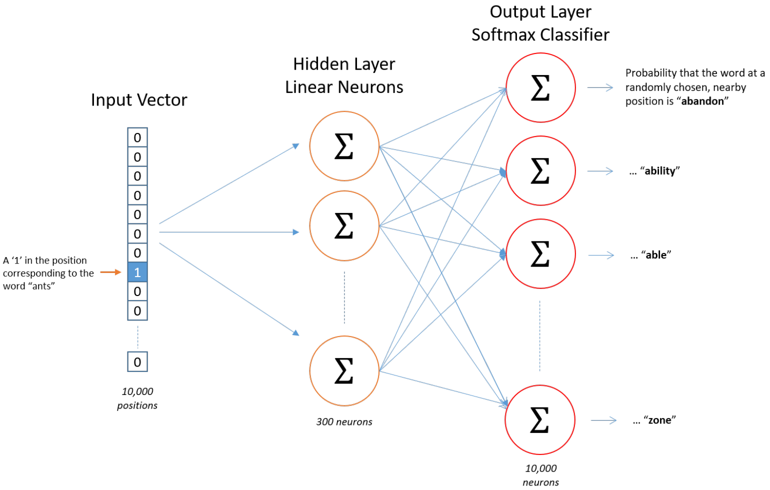
Word Embeddings



Word Embeddings Algorithms

- Word embeddings are build by training **neural networks** architectures on document **corpora** (e.g., books, papers, Wikipedia, tweets, the Web).
- These arquitectures formulate a **predictive task** (e.g., predict a missing word withing a contex window) in which word embeddings naturally arise from the network's parameters after training.
- Most popular models are:
  - Skip-gram negative sampling [Mikolov et al., 2013]
  - Continuous bag-of-words [Mikolov et al., 2013]
  - Glove [Pennington et al., 2014]
  - FastText [Bojanowski et al., 2016].

Skip-gram Model



Picture taken from: `http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/`

Limitations of Word Embeddings

- In this talk we will present our research addressing three limitations of word embeddings.
  1. **Fairness**: they are prone to inherit stereotypical social biases from the corpus they were built on.
  2. **Change**: they are static. Thus they ignore words not observed during training and are unable to capture semantic drifts.
  3. **Polysemy**: they fail to capture the polysemous nature of many words (e.g., apple:company, apple:fruit), conflating their multiple senses into a single point.

Fairness in Word Embeddings

Word Embeddings have showed to exhibit undesireded associations between words for certain social groups (e.g., gender, race, religion):

- Black is to criminal as caucasian is to police.
- Man is to doctor as woman is to nurse.
- Man is to computer programmer as woman is to homemaker.

Fairness Metrics

- **WEAT** [Caliskan et al., 2017]: Degree of association between two sets of target words and two sets of attribute words.
  - Studies biases regarding ethnicity (in relation to pleasant-unpleasant words) and gender (in relation to occupations).
- **RND** [Garg et al., 2018]: Compares the embeddings related to two different social group against a set of neutral words.
  - Embeddings trained on different periods of time were evaluated to test to gender and ethnic biases.
- **RNSB** [Sweeney and Najafian, 2019]: KL divergence between the negative sentiment probability of the embeddings and a uniform distribution.
  - Test a set of national origin identity terms such as American, Mexican, and Canadian.

WEFE: The Word Embeddings Fairness Evaluation Framework

- Although metrics have similar objectives they cannot be easily compared with each other.
- They operate with different inputs (e.g., pairs of words, sets of words, multiple sets of words).
- Their outputs are also incompatible with each other (e.g., reals, positive numbers, $[0, 1]$ range).
- The **Word Embeddings Fairness Evaluation (WEFE)** is a framework for measuring and mitigating bias in word embeddings [Badilla et al., 2020].
- WEFE formalizes existing fairness metrics into a **unified framework**.

WEFE: The Word Embeddings Fairness Evaluation Framework

- It encapsulates the input words of these metrics into standard objects called **queries**.
- Then, proceeds to quantify the bias of word embeddings using these queries along with different fairness metrics.
- It also allows for **ranking** multiple embeddings models by aggregating the previous scores.

WEFE Building Blocks

- **Target Sets**: a set of words intended to denote a particular social group. For example:

$$T_{\text{women}} = \{w_{\text{she}}, w_{\text{woman}}, w_{\text{girl}}, \ldots\}, \quad (1)$$
$$T_{\text{men}} = \{w_{\text{he}}, w_{\text{man}}, w_{\text{boy}}, \ldots\}, \quad (2)$$

- **Attribute Sets**: a set of words representing some attitude, characteristic, trait, occupational field, etc. that can be associated with individuals from any social group.
  For example:

$$A_{\text{science}} = \{w_{\text{math}}, w_{\text{physics}}, w_{\text{chemistry}}, \ldots\}, \quad (3)$$
$$A_{\text{art}} = \{w_{\text{poetry}}, w_{\text{dance}}, w_{\text{literature}}, \ldots\}. \quad (4)$$

- **Query**: a pair $Q = (\mathcal{T}, \mathcal{A})$ in which $\mathcal{T}$ is a set of target word sets, and $\mathcal{A}$ is a set of attribute word sets.
  For example:

$$Q = (\{T_{\text{women}}, T_{\text{men}}\}, \{A_{\text{science}}, A_{\text{art}}\}). \quad (5)$$

WEFE Building Blocks

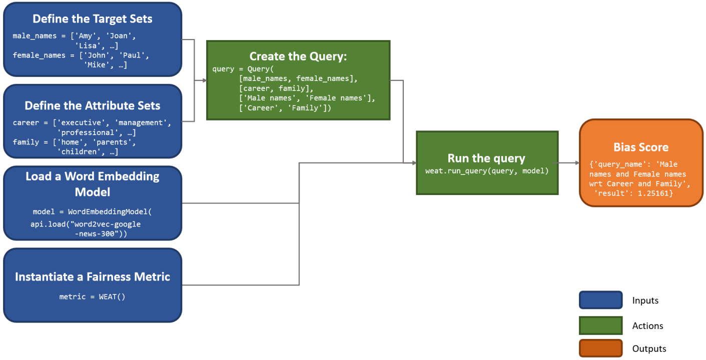
- Queries are the main building blocks used by fairness metrics to measure bias of word embedding models.
- Fairness metrics encapsulated by WEFE are defined with a query template $s_F = (t, a)$, which is a pair of natural values with the cardinalities of sets $\mathcal{T}$ and $\mathcal{A}$.
- We equip each metric with a total order relation $\leq_F$ that establishes what is to be considered as *less biased*.
- That is, if

$$F(\mathbf{M_1}, Q) \leq_F F(\mathbf{M_2}, Q)$$

  we can say that $\mathbf{M_1}$ *is less biased than model* $\mathbf{M_2}$.
- With this order relation we can rank various embeddings models according to their biases.

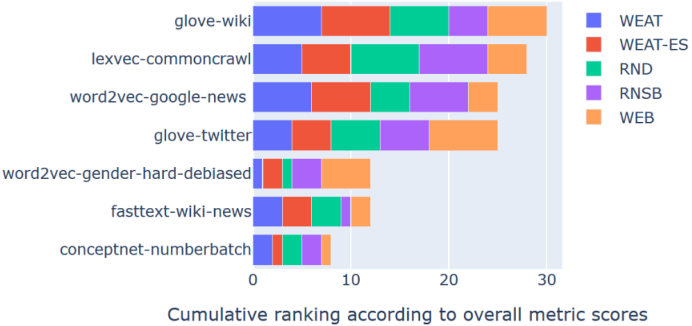
WEFE Toolkit

We released WEFE as an open source Python software:
`https://wefe.readthedocs.io/`



Case Study

We used WEFE to rank various embedding models according to their aggregated biased with respect to to gender, religion and ethnicity:

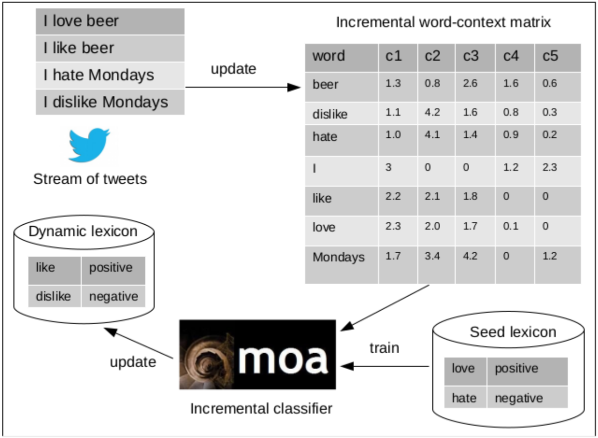


Incremental Word Vectors for time-evolving sentiment lexicon induction

- A sentiment lexicon is a list of expressions annotated according to affect categories such as positive, negative, anger and fear.
- Lexicons are widely used in sentiment classification of tweets.
- They are usually build by training a word-level sentiment classifier on pre-trained word embeddings using a **seed lexicon** as training data [Tang et al., 2014].
- However, as word embeddings are static by nature, sentiment lexicons build in this way are unable to capture:
    1. The arrival of new sentiment-conveying expressions such as #trumpwall and #PrayForParis.
    2. Temporal changes in sentiment patterns of words (e.g., a scandal associated with an entity).

Incremental Word Vectors for time-evolving sentiment lexicon induction

We developed a methodology for automatically inducing **continuously updated** sentiment lexicons from **Twitter streams** by training **incremental** word sentiment classifiers from **time-evolving word vectors**.
[Bravo-Marquez et al., 2021]



Incremental Word Vectors for time-evolving sentiment lexicon induction

The process is summarized as follows:

1. Connect to a stream of continuously arriving text (e.g., Twitter).
2. Every time a new tweet arrives, a sliding window of W words centered on a target word is shifted across the content.
3. The word vector associated with the target word is updated according to its context. This process implies updating word-context counts and computing PPMI-based associations.
4. If the target word is new, a new vector associated with this word is created.
5. If the target word is contained in the seed lexicon, the incremental classifier is updated/trained using the target word vector and the lexicon's sentiment as the gold label.
6. If the target word is not contained in the lexicon, its sentiment is estimated using the classifier and the dynamic lexicon is updated.

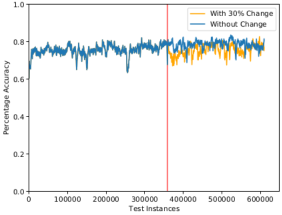
Incremental Algorithms

Data: tweets, window size W, vocabulary size V, context size C
Result: PPMI Matrix M
Initialize PPMI Matrix M of size $V \times C$;
$D \leftarrow 0$;
while *tweet in Data Stream* **do**
  tokens $\leftarrow$ tokenize(tweet);
  for *each* *w in tokens* **do**
    $D \leftarrow D + 1$;
    $c_1, \ldots, c_{2W} \leftarrow$ getContexts(w,tokens);
    updateContextMatrix($w, c_1, \ldots, c_{2W}$);
    for *each* *c_j in $c_1, \ldots, c_{2W}$* **do**
      PPMI$(w, c_j) \leftarrow$
      $\max\left(0, \log_2\left(\frac{count(w,c_j) \times D}{count(w) \times count(c_j)}\right)\right)$;
    end
  end
end

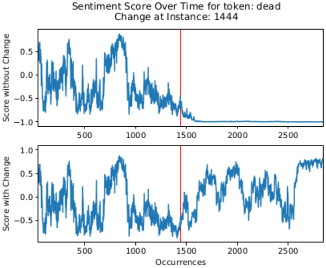
Data: tweets, training seed lexicon, testing seed lexicon
while *Tweet T in Data Stream* **do**
  for *each* *token t in tokens* **do**
    updateContextCounters(t,T);
    updatePPMIValues(t,T);
    if *token $\in$ train seed lexicon* **then**
      trainClassifier(PPMI(t),label(t));
    else if *token $\in$ test seed lexicon* **then**
      updateEvaluator(PPMI(t),label(t));
  end
end

Sentiment Drifts

- We simulate sentiment change by randomly picking some words and swapping their context with the context of words exhibiting the opposite sentiment.



1. Accuracy for experiment with 30% of Change

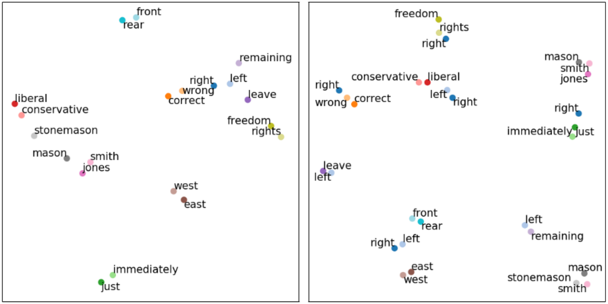


1. (a) dead

- Our approach allows for successfully tracking of the sentiment of words over time even when drastic change is induced.

PolyLM: a polysemous language model

- PolyLM is a language model capable of automatically learning multiple meanings of a word (e.g. apple:apple, apple:company) [Ansell et al., 2021].



(a) Word embeddings (b) Sense embeddings

- The occurrence of closely related polysemous words nearby in the word embedding space (i.e. *left* and *right*) causes unrelated words to be closer together (e.g. *left* and *wrong*) and related words to be further apart (e.g. *right* and *east*) than they otherwise would be

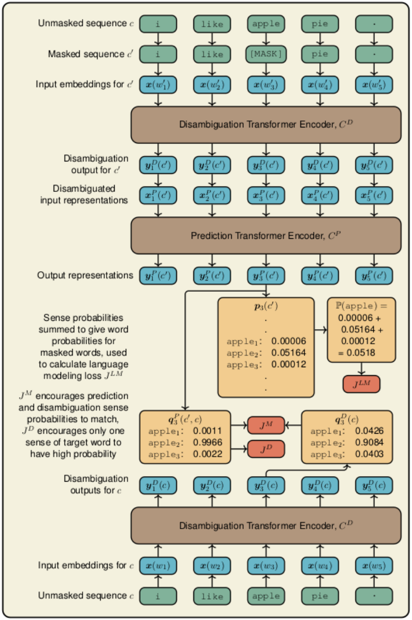
PolyLM: a polysemous language model

- PolyLM is an unsupervised language model based on two assumptions:
  1. The probability of a word occurring in a given context is equal to the sum of the probabilities of its individual senses.

$$p(w|c) = \sum_{s \in S_w} p(s|c). \tag{6}$$

  2. For a given occurrence of a word, one of its senses tends to be much more plausible in the context than the others.
- PolyLM is based on a Transformer encoder architecture and the masked language modeling (MLM) task used for training BERT.
- Its loss function combines a standard language modeling loss with a distinctness loss encouraging one sense to have a high estimated probability of occurring, and the rest to have a low probability.

PolyLM: a polysemous language model



Results on the Word Sense Induction (WSI) task

System	Version	SemEval-2010			SemEval-2013		
		F-S	V-M	**AVG**	FBC	FNMI	**AVG**
Aamrami-goldberg	$\text{BERT}_{\text{LARGE}}$	**71.3**	40.4	**53.6**	64.0	21.4	37.0
AutoSense		62.9	10.1	25.2	61.7	8.0	22.2
PolyLM†	BASE	65.8	**40.5**	51.6	**64.8**	**23.0**	**38.6**
	SMALL	65.6	35.7	48.4	64.5	18.5	34.5
Qiu et. al†		-	-	-	56.9	6.7	19.5
SE-WSI-fix-cmp†		54.3	16.3	29.8	-	-	-
AdaGram†		43.9	20.0	29.6	13.2	8.9	10.8
Arora et. al†	$k = 5$	46.4	11.5	23.1	-	-	-

Table: Comparison of sense embedding models and WSI-specific techniques on the SemEval 2010 and 2013 WSI tasks. SE-WSI-fix-cmp is based on MSSG model. † - models which obtain explicit sense embeddings.

Conclusions

- In this talk we have presented three research projects, each addressing a different limitation of word embeddings.
- I would like to thank all the co-authors of the presented works: Alan, Arun, Jorge, Pablo and Bernhard.
- There is plenty of room for further research.

Questions?

Thanks for your Attention!

References I

Ansell, A., Bravo-Marquez, F., and Pfahringer, B. (2021).
Polylm: Learning about polysemy through language modeling.
In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 563–574.

Badilla, P., Bravo-Marquez, F., and Pérez, J. (2020).
Wefe: The word embeddings fairness evaluation framework.
In *IJCAI*, pages 430–436.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016).
Enriching word vectors with subword information.
arXiv preprint arXiv:1607.04606.

Bravo-Marquez, F., Khanchandani, A., and Pfahringer, B. (2021).
Incremental word vectors for time-evolving sentiment lexicon induction.
Cognitive Computation, pages 1–17.

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017).
Semantics derived automatically from language corpora contain human-like biases.
Science, 356(6334):183–186.

Garg, N., Schiebinger, L., Jurafsky, D., and Zou, J. (2018).
Word embeddings quantify 100 years of gender and ethnic stereotypes.
Proceedings of the National Academy of Sciences, 115(16):E3635–E3644.

References II

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013).
Distributed representations of words and phrases and their compositionality.
In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Pennington, J., Socher, R., and Manning, C. D. (2014).
Glove: Global vectors for word representation.
In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Sweeney, C. and Najafian, M. (2019).
A transparent framework for evaluating unintended demographic bias in word embeddings.
In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1662–1667.

References III

Tang, D., Wei, F., Qin, B., Zhou, M., and Liu, T. (2014).
Building large-scale twitter-specific sentiment lexicon : A representation learning approach.
In *Proceedings of the 25th International Conference on Computational Linguisticss*, pages 172–182. Association for Computational Linguistics.