Embeddings Words and Senses Together via Joint Knowledge-Enhanced Training

Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci and Roberto Navigli

lcl.uniroma1.it/sw2v
Motivation: Model senses instead of only words

*He withdrew money from the bank.*
Motivation: Model senses instead of only words

*He withdrew money from the bank.*
Motivation: Model senses instead of only words

He withdrew money from the bank.
Related Work

➢ Unsupervised sense embeddings

➢ Knowledge-based sense embeddings
Related Work

➢ Unsupervised sense embeddings

Learn sense embeddings exploiting text corpora only (Huang et al. ACL 2012; Neelakantan et al. EMNLP 2014; Tian et al. COLING 2014; Li and Jurafsky, EMNLP 2015...). Easily adaptable to new domains.

Drawbacks:

- Senses not interpretable (+change from model to model)
- Knowledge from resources cannot be easily exploited
- Senses (esp. not frequent ones) not easy to discriminate

➢ Knowledge-based sense embeddings
Related Work

➢ Unsupervised sense embeddings

➢ Knowledge-based sense embeddings

Model senses as defined on a sense inventory.

Usually obtained as a postprocessing of word embeddings (Chen et al. EMNLP 2014; Rothe and Schütze, ACL 2015…):

- Several training phases
- Infrequent senses not accurately captured
Related Work

➢ Unsupervised sense embeddings

➢ Knowledge-based sense embeddings (Our approach)
Related Work

➢ Unsupervised sense embeddings

➢ Knowledge-based sense embeddings (Our approach)
A word is the surface form of a sense: we can exploit this intrinsic relationship for jointly training word and sense embeddings.
Idea

A word is the surface form of a sense: we can exploit this intrinsic relationship for **jointly training word and sense embeddings**.

**How?**

Updating the representation of the word and its associated senses interchangeably.
Methodology

Given as input a corpus and a semantic network:

1. Use a semantic network to link to each word its associated senses in context.

   *He withdrew money from the bank.*
Given as input a **corpus** and a **semantic network**:

1. Use a semantic network to link to each word its associated senses *in context*.

*He withdrew money from the bank.*
Methodology: Linking words and senses in context

He **withdrew** money from the **bank**

- *retire*
- *cash*
- *take out*
- *geography*
- *financial institution*
- *building*
He withdrew money from the bank.

Graph-based representation of the sentence using semantic networks (e.g. WordNet, BabelNet)
He withdrew money from the bank

Graph-based representation of the sentence using semantic networks (e.g. WordNet, BabelNet)
Methodology

Given as input a corpus and a semantic network:

1. Use a semantic network to link to each word its associated senses in context.

2. Use a neural network where the update of word and sense embeddings is linked, exploiting virtual connections.
Methodology

Given as input a corpus and a semantic network:

1. Use a semantic network to link to each word its associated senses in context.

2. Use a neural network where the update of word and sense embeddings is linked, exploiting virtual connections.
Methodology

Given as input a corpus and a semantic network:

1. Use a semantic network to link to each word its associated senses in context.

2. Use a neural network where the update of word and sense embeddings is linked, exploiting *virtual* connections.
Methodology

Given as input a corpus and a semantic network:

1. Use a semantic network to link to each word its associated senses in context.

2. Use a neural network where the update of word and sense embeddings is linked, exploiting virtual connections.
Methodology

Given as input a corpus and a semantic network:

1. Use a semantic network to link to each word its associated senses in context.

2. Use a neural network where the update of word and sense embeddings is linked, exploiting virtual connections.
Methodology

Given as input a corpus and a semantic network:

1. Use a semantic network to link to each word its associated senses in context.

2. Use a neural network where the update of word and sense embeddings is linked, exploiting virtual connections.
Methodology

Given as input a corpus and a semantic network:

1. Use a semantic network to link to each word its associated senses in context.

2. Use a neural network where the update of word and sense embeddings is linked, exploiting virtual connections.

In this way it is possible to learn word and sense/synset embeddings jointly on a single training.
Methodology: Joint training of words and sense embeddings

Once each word is connected to its set of senses in context, it is possible to modify standard word embedding architectures to take into account this information.

In this work we explore the CBOW architecture of Word2Vec (Mikolov et al. 2013) -> **SW2V (Senses and Words to Vectors)**.

Other neural network architectures could be explored as well (Skip-gram also included in the code).
Full architecture of W2V (Mikolov et al. 2013)

\[ E = -\log(p(w_t|W^t)) \]

Words and associated senses used both as input and output.
Full architecture of SW2V (this work)

$$E = -\log(p(w_t|W^t, S^t)) - \sum_{s \in S^t} \log(p(s|W^t, S^t))$$

Words and associated senses used both as input and output.
Output layer alternatives: only words

\[ E = -\log(p(w_t|W^t, S^t)) - \sum_{s \in S^t} \log(p(s|W^t, S^t)) \]

The architecture does not try to predict senses. No loss contribution from them.
The architecture does not try to predict words. No loss contribution from them.
Input layer alternatives: only words

\[ E = -\log(p(w_t|W^t, S^t)) - \sum_{s \in S^t} \log(p(s|W^t, S^t)) \]

Senses are not included in the input layer. Only words contribute to the hidden state. This way, during backpropagation sense embeddings do not receive any gradient.
During backpropagation, sense embeddings will receive the same gradient of the word they are associated with.

\[ E = -\log(p(w_t|W_t, S_t)) - \sum_{s \in S_t} \log(p(s|W_t, S_t)) \]
Input layer alternatives: only senses

$E = -\log(p(w_t|W^t, S^t)) - \sum_{s \in S^t} \log(p(s|W^t, S^t))$

Words are not included in the input layer. Only senses contribute to the hidden state. This way, during backpropagation word embeddings do not receive any gradient.
Input layer alternatives: only senses

\[ E = -\log(p(w_t|W^t, S^t)) - \sum_{s \in S^t} \log(p(s|W^t, S^t)) \]

During backpropagation, their embeddings will receive the **same** gradient of their associated senses.
Analysis: Model configurations

We used word similarity for analyzing the performance of sense embeddings on each of the nine configurations.

- Best configuration -

- Input layer: Only senses
- Output layer: Both words and senses

Why? (Intuition) Co-occurrence information gets duplicated if both words and senses are included in the input layer.
Evaluation: Experimental setting

➢ Best configuration used in all experiments
➢ Standard hyperparameters
➢ Semantic networks used: WordNet and BabelNet
➢ Corpora used: UMBC and Wikipedia
➢ Experiments on:
  - Word and sense interconnectivity (qualitative)
  - Word similarity
  - Sense clustering
Evaluation: Comparison systems

Sense embeddings:

➢ Chen et al. (2014)

⭐ ➢ AutoExtend (Rothe and Schütze, 2015)

➢ SensEmbed (Iacobacci et al. 2015)

➢ NASARI (Camacho-Collados et al. 2016)
Evaluation: Comparison systems

Sense embeddings:

- Chen et al. (2014)
- AutoExtend (Rothe and Schütze, 2015)
- SensEmbed (Iacobacci et al. 2015)
- NASARI (Camacho-Collados et al. 2016)
Evaluation: Comparison systems

Sense embeddings:

➢ Chen et al. (2014)
➢ AutoExtend (Rothe and Schütze, 2015)
➢ SensEmbed (Iacobacci et al. 2015)
➢ NASARI (Camacho-Collados et al. 2016)

Word embeddings:

➢ Word2Vec (Mikolov et al. 2013)
★ ➢ Retrofitting (Faruqui et al. 2015)
Evaluation: Comparison systems

Sense embeddings:

➢ Chen et al. (2014)
➢ AutoExtend (Rothe and Schütze, 2015)
➢ SensEmbed (Iacobacci et al. 2015)
➢ NASARI (Camacho-Collados et al. 2016)

Word embeddings:

➢ Word2Vec (Mikolov et al. 2013)
➢ Retrofitting (Faruqui et al. 2015)
Evaluation: Word and sense interconnectivity

How coherent is the shared vector space of word and sense embeddings?

**Intuition:** the Most Frequent Sense (MFS) should be close to the word embedding -> Reasonably strong MFS baseline for WSD

Evaluation on two WSD datasets using the **embeddings as a MFS baseline** (closest sense embedding to its associated word embedding is selected).
Evaluation: Word and sense interconnectivity

Embedding Words and Senses Together via Joint Knowledge-Enhanced training
Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci and Roberto Navigli
Word and sense interconnectivity: Example I

Ten closest word and sense embeddings to the sense *company* (military unit)
Word and sense interconnectivity: Example II

Ten closest word and sense embeddings to the sense school (group of fish)
Evaluation: Word similarity

All models using Wikipedia corpus (Pearson correlation)
Evaluation: Word similarity

All models using Wikipedia corpus (Pearson correlation)
Evaluation: Word similarity

All models using UMBC corpus (Pearson correlation)
Evaluation: Sense clustering

Some sense inventories make a fine-grained distinction between senses, which can be harmful on downstream applications (Hovy et al. 2013, Pilehvar et al. 2017).

**Example:** *Bank*  
- *Institution*
- *Physical building*

**Evaluation datasets** (Dandala et al. 2013): Highly ambiguous words from past SemEval competitions.
Evaluation: Sense clustering

![Bar chart showing accuracy and F-Measure for different methods: SW2V, NASARI, SensEmbed, Multi-SVM, Mono-SVM. The accuracy and F-Measure values are as follows:

- **Accuracy**: SW2V 87.8, NASARI 87, SensEmbed 82.7, Multi-SVM 85.5, Mono-SVM 83.5
- **F-Measure**: SW2V 63.9, NASARI 62.5, SensEmbed 40.3

Embedding Words and Senses Together via Joint Knowledge-Enhanced training
Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli
Conclusion

We presented SW2V: a neural architecture for **jointly learning word and sense embeddings** in the same vector space using text corpora and knowledge obtained from semantic networks.
Conclusion

We presented SW2V: a neural architecture for jointly learning word and sense embeddings in the same vector space using text corpora and knowledge obtained from semantic networks.

Future work:
- Exploiting our model for other linked representations such as multilingual or Image-to-Text embeddings.
- Word Sense Disambiguation and Entity Linking.
- Integrating our embeddings into downstream NLP applications, following the lines of Pilehvar et al. (ACL 2017).
Conclusion

We presented SW2V: a neural architecture for jointly learning word and sense embeddings in the same vector space using text corpora and knowledge obtained from semantic networks.

Future work:

- Exploiting our model for other linked representations such as multilingual or Image-to-Text embeddings.
- Word Sense Disambiguation and Entity Linking.
- Integrating our embeddings into downstream NLP applications, following the lines of Pilehvar et al. (ACL 2017).

http://lcl.uniroma1.it/sw2v
Thank you!

Code and pre-trained models available at

http://lcl.uniroma1.it/sw2v
SECRET SLIDES
Outline

➢ Related work

➢ Our approach: SW2V (*Senses and Words to Vectors*)
  ○ Linking words and senses in context
  ○ Joint training of words and sense embeddings

➢ Evaluation
Methodology

Given as input a corpus and a semantic network:

1. Use a semantic network to link to each word its associated senses in context.

*He withdrew money from the bank.*
Joint training of word and sense embeddings

Once each word is connected to its set of senses in context, it is possible to modify standard word embedding models to take into account this information.

Formally, given a target word at position $t$ we have a set of words:

$$W = \{w_{t-n}, \ldots, w_t, \ldots, w_{t+n}\} \quad \text{with} \quad W^t = W \setminus w_t$$

and a set of associated senses:

$$S = \{S_{t-n}, \ldots, S_t, \ldots, S_{t+n}\} \quad \text{and} \quad S^t = S \setminus S_t$$

with

$$S_i = \{s_{i,1}, \ldots, s_{i,k,i}\}$$

the senses associated with the $i_{th}$ word.

We aim at minimizing:

$$E = -\log(p(w_t | W^t, S^t)) - \sum_{s \in S_t} \log(p(s | W^t, S^t))$$
## Evaluation: Word similarity

<table>
<thead>
<tr>
<th>Sense Embeddings</th>
<th>SimLex-999</th>
<th>MEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Corpus</td>
<td>r</td>
</tr>
<tr>
<td>SW2V&lt;sub&gt;BN&lt;/sub&gt;</td>
<td>UMBC</td>
<td>0.49</td>
</tr>
<tr>
<td>SW2V&lt;sub&gt;WN&lt;/sub&gt;</td>
<td>UMBC</td>
<td>0.46</td>
</tr>
<tr>
<td>AutoExtend</td>
<td>UMBC</td>
<td>0.47</td>
</tr>
<tr>
<td>AutoExtend</td>
<td>Google-News</td>
<td>0.46</td>
</tr>
<tr>
<td>SW2V&lt;sub&gt;BN&lt;/sub&gt;</td>
<td>Wikipedia</td>
<td>0.47</td>
</tr>
<tr>
<td>SW2V&lt;sub&gt;WN&lt;/sub&gt;</td>
<td>Wikipedia</td>
<td>0.47</td>
</tr>
<tr>
<td>SensEmbed</td>
<td>Wikipedia</td>
<td>0.43</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word Embeddings</th>
<th>SimLex-999</th>
<th>MEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Corpus</td>
<td>r</td>
</tr>
<tr>
<td>Word2Vec</td>
<td>UMBC</td>
<td>0.39</td>
</tr>
<tr>
<td>Retrofitting&lt;sub&gt;BN&lt;/sub&gt;</td>
<td>UMBC</td>
<td>0.47</td>
</tr>
<tr>
<td>Retrofitting&lt;sub&gt;WN&lt;/sub&gt;</td>
<td>UMBC</td>
<td>0.47</td>
</tr>
<tr>
<td>Word2Vec</td>
<td>Wikipedia</td>
<td>0.39</td>
</tr>
<tr>
<td>Retrofitting&lt;sub&gt;BN&lt;/sub&gt;</td>
<td>Wikipedia</td>
<td>0.35</td>
</tr>
<tr>
<td>Retrofitting&lt;sub&gt;WN&lt;/sub&gt;</td>
<td>Wikipedia</td>
<td>0.47</td>
</tr>
</tbody>
</table>

*Embedding Words and Senses Together via Joint Knowledge-Enhanced training*

Massimiliano Mancini, **Jose Camacho-Collados**, Ignacio Iacobacci and Roberto Navigli
Evaluation: Word similarity

Wikipedia: Pearson correlation

Embedding Words and Senses Together via Joint Knowledge-Enhanced training
Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci and Roberto Navigli
Evaluation: Word similarity

Wikipedia: Pearson correlation

Embedding Words and Senses Together via Joint Knowledge-Enhanced training
Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci and Roberto Navigli
Evaluation: Word similarity

UMBC: Pearson correlation

Embedding Words and Senses Together via Joint Knowledge-Enhanced training
Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci and RobertoNavigli
Evaluation: Word similarity

Wikipedia: Spearman correlation

Embedding Words and Senses Together via Joint Knowledge-Enhanced training
Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci and Roberto Navigli
Evaluation: Word similarity

UMBC: Spearman correlation

Embedding Words and Senses Together via Joint Knowledge-Enhanced training
Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci and Roberto Navigli
### Evaluation: Sense clustering

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>17.5</td>
<td>29.8</td>
</tr>
<tr>
<td>Multi-SVM</td>
<td>85.5</td>
<td>62.5</td>
</tr>
<tr>
<td>Mono-SVM</td>
<td>83.5</td>
<td>63.9</td>
</tr>
<tr>
<td>SensEmbed</td>
<td>82.7</td>
<td>87</td>
</tr>
<tr>
<td>NASARI</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>SW2V</td>
<td>87.8</td>
<td>63.9</td>
</tr>
</tbody>
</table>

Embedding Words and Senses Together via Joint Knowledge-Enhanced training
Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci and Roberto Navigli
## Evaluation: Sense clustering

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW2V</td>
<td>87.8</td>
<td>63.9</td>
</tr>
<tr>
<td>SensEmbed</td>
<td>82.7</td>
<td>40.3</td>
</tr>
<tr>
<td>NASARI</td>
<td>87.0</td>
<td>62.5</td>
</tr>
<tr>
<td>Multi-SVM</td>
<td>85.5</td>
<td>-</td>
</tr>
<tr>
<td>Mono-SVM</td>
<td>83.5</td>
<td>-</td>
</tr>
<tr>
<td>Baseline</td>
<td>17.5</td>
<td>29.8</td>
</tr>
</tbody>
</table>

**Embedding Words and Senses Together via Joint Knowledge-Enhanced training**

Massimiliano Mancini, *Jose Camacho-Collados*, Ignacio Iacobacci and Roberto Navigli
# Word and sense interconnectivity

<table>
<thead>
<tr>
<th></th>
<th>SemEval-07</th>
<th>SemEval-13</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW2V</td>
<td>39.9</td>
<td>54.0</td>
</tr>
<tr>
<td>AutoExtend</td>
<td>17.6</td>
<td>31.0</td>
</tr>
<tr>
<td>Baseline</td>
<td>24.8</td>
<td>34.9</td>
</tr>
</tbody>
</table>