Embedding Words and Senses Together via Joint Knowledge-Enhanced Training

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

Table of Contents

Introduction

- Popularity of approaches based on neural networks which embed words into low-dimensional vector spaces from text corpora (word embeddings)
- Word embeddings in Natural Language Processing tasks:
  - Machine Translation
  - Syntactic Parsing
  - Question Answering
  - Many more
- Limitation of embeddings: the inability to discriminate among different meanings of the same word

Introduction

- Previous works have tried to address this word sense disambiguation limitation by automatically identifying word senses from monolingual corpora or bilingual parallel data
- Problems with these approaches:
  - Learn solely on the basis of statistics extracted from text corpora and do not use semantic networks
  - Identified senses are not readily interpretable or easily mappable to lexical resources

Introduction

- Recent approaches utilize semantic networks to inject knowledge into existing word representations, but cannot determine the correct senses
- Other approaches leverage lexical resources to learn sense embeddings as a result of post-processing conventional word embeddings in order to obtain a representation for each sense of the word

Introduction

- SW2V: Senses and Words to Vectors
  - A neural model that exploits knowledge from both text corpora and semantic networks in order to simultaneously learn embeddings for both words and senses
  - Both word and sense embeddings are represented in the same vector space
  - Flexible, so can be applied to different predictive models
  - Scalable for very large semantic networks and text corpora

Related Work

- Embedding words from large corpora into a low-dimensional vector space has been a popular task
  - Probabilistic feed-forward neural network language model
  - Word2vec
  - GloVe
- Little research has focused on exploiting lexical resources to overcome the inherent ambiguity of word embeddings

Related Work

- One approach: apply an off-the-shelf disambiguation system to a corpus and then using word2vec to learn sense embeddings over the pre-disambiguated text
- Problem with this approach: words are replaced by their intended senses, consequently producing as output sense representations only
- The representation of words and senses in the same vector space is essential for applying these knowledge based sense embeddings in downstream applications, particularly for their integration into neural architectures

Related Work

- WordNet: a model for obtaining both word and sense representations based on a first training step of conventional word embeddings, a second disambiguation step based on sense definitions, and a final training phase which uses the disambiguated text as input
- Building a shared space of word and sense embeddings based on two steps: a first training step of only word embeddings and a second training step to produce sense and synset embeddings
- Above approaches require multiple steps of training and make use of a relatively small resource (WordNet), which limits their coverage and applicability

Related Work

- Wikipedia: increasing the coverage of the WordNet based approaches by exploiting the complementary knowledge of WordNet and Wikipedia along with pre-trained word embeddings
- A model to align vector spaces of words and entities from knowledge bases
- Limitations: these approaches are restricted to nominal instances only (Wikipedia pages or entities)
- In contrast, the authors propose a model that learns both words and sense embeddings from a single joint training phase, producing a common vector space of words and senses as an emerging feature

Table of Contents

Shallow Word-Sense Connectivity Algorithm

- SW2V needs a corpus where words are connected to senses in each given context
- Options: manually annotating corpora
  - Expensive, therefore impractical in normal settings
- Data from current off-the-shelf disambiguation and entity linking systems
  - Supervised systems need large amounts of sense-annotated data
  - Slow speed- hampers progress when applied to large corpora
- Shallow word-sense connectivity algorithm
  - Can be applied to virtually any given network
  - Exploit the connects of a semantic network by associating words with the senses that are most connected within the sentence, according to the underlying network

Shallow Word-Sense Connectivity Algorithm

- A corpus and a semantic network are taken as input and a set of connected words and sense is produced as output
- Semantic network: a graph (S, E)
  - The set S contains synsets (nodes)
  - E represents a set of semantically connected synset pairs (edges)

Shallow Word-Sense Connectivity Algorithm

Algorithm 1 Shallow word-sense connectivity

Input: Semantic network (S, E) and text T represented as a bag of words

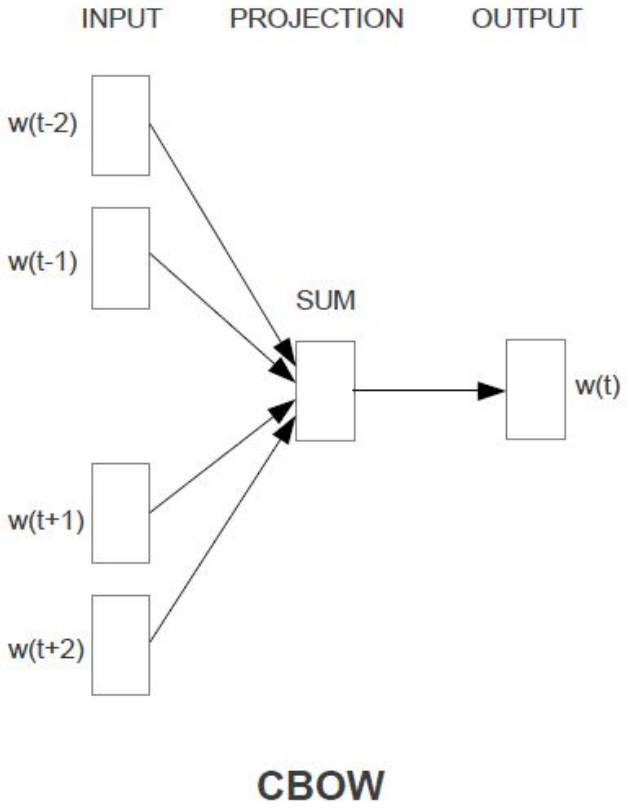
Output: Set of connected words and senses $T^* \subset T \times S$

Set of synsets $S_T \leftarrow \emptyset$
for each word $w \in T$
    $S_T \leftarrow S_T \cup S_w$ (S_w: set of candidate synsets of w)
Minimum connections threshold $\theta \leftarrow \frac{|S_T|+|T|}{2\,\delta}$
Output set of connections $T^* \leftarrow \emptyset$
for each $w \in T$
    Relative maximum connections $max = 0$
    Set of senses associated with w, $C_w \leftarrow \emptyset$
    for each candidate synset $s \in S_w$
        Number of edges $n = |s' \in S_T : (s, s') \in E \;\; \& \;\; \exists w' \in T : w' \neq w \;\; \& \;\; s' \in S_{w'}|$
        if $n \geq max \;\; \& \;\; n \geq \theta$ **then**
            if $n > max$ **then**
                $C_w \leftarrow \{(w, s)\}$
                $max \leftarrow n$
            else
                $C_w \leftarrow C_w \cup \{(w, s)\}$
    $T^* \leftarrow T^* \cup C_w$
return Output set of connected words and senses T^*

Table of Contents

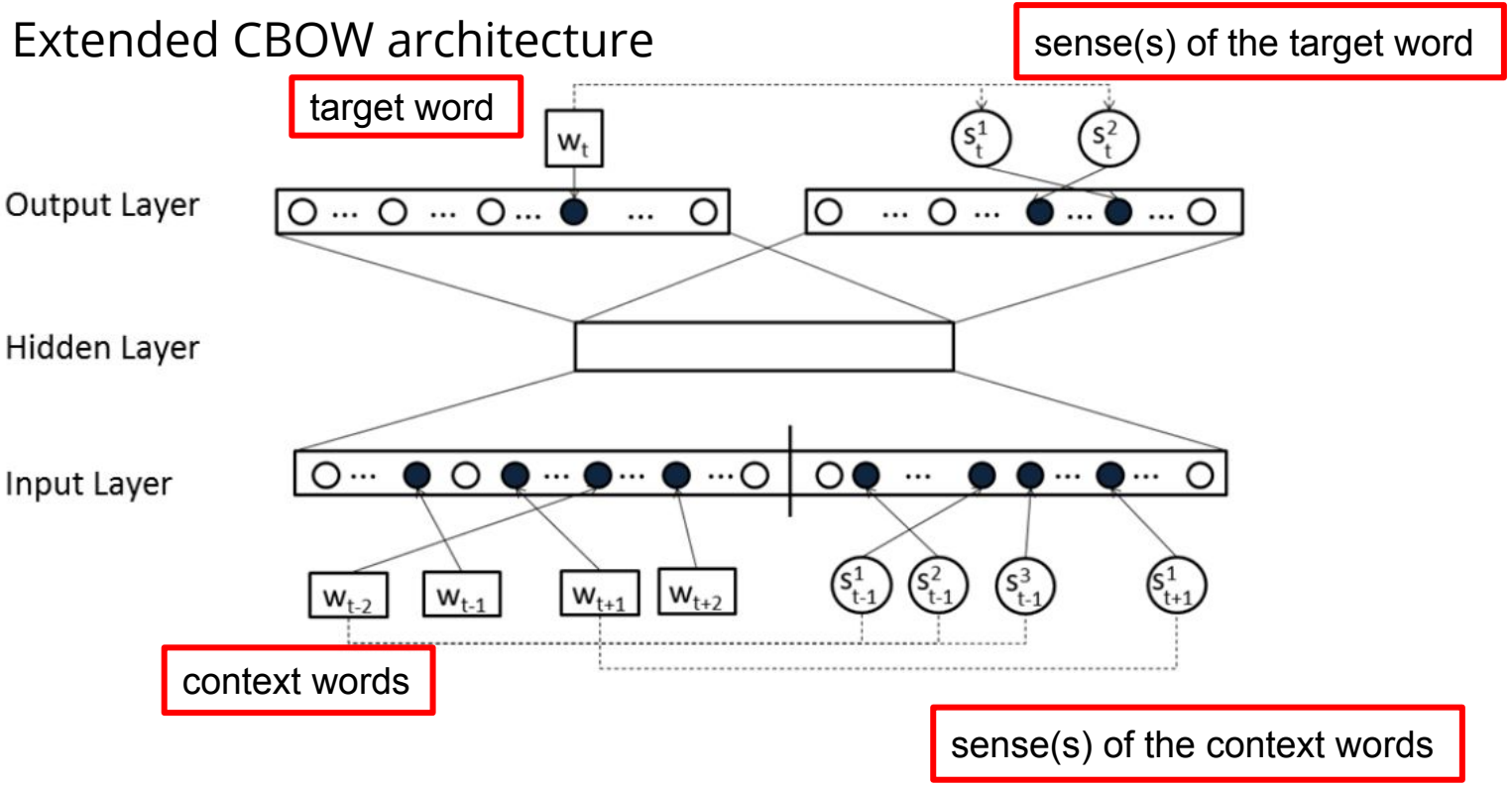
Joint Training of words and senses

- Goal: shared vector space for words and senses
- Extended CBOW architecture



Joint Training of words and senses

- Extended CBOW architecture



Joint Training of words and senses

- Goal: shared vector space of words and senses
- Extended CBOW architecture
  - Other predictive frameworks would work as well
- Why does this make sense?
  - Intrinsic relationship between words and senses
  - Updating representation of one consequently produces update of the other

Joint Training of words and senses

- Formal architecture:

Training instance:

target

$$\begin{cases} W^t = w_{t-n}, \ldots, \boxed{w_n}, \ldots, w_{t+n} \\ S^t = S_{t-n}, \ldots, \boxed{S_n}, \ldots, S_{t+n} \end{cases}$$

$S_i = s^1_i, \ldots, s^k_i$ Sequence of associated senses in context of w_i

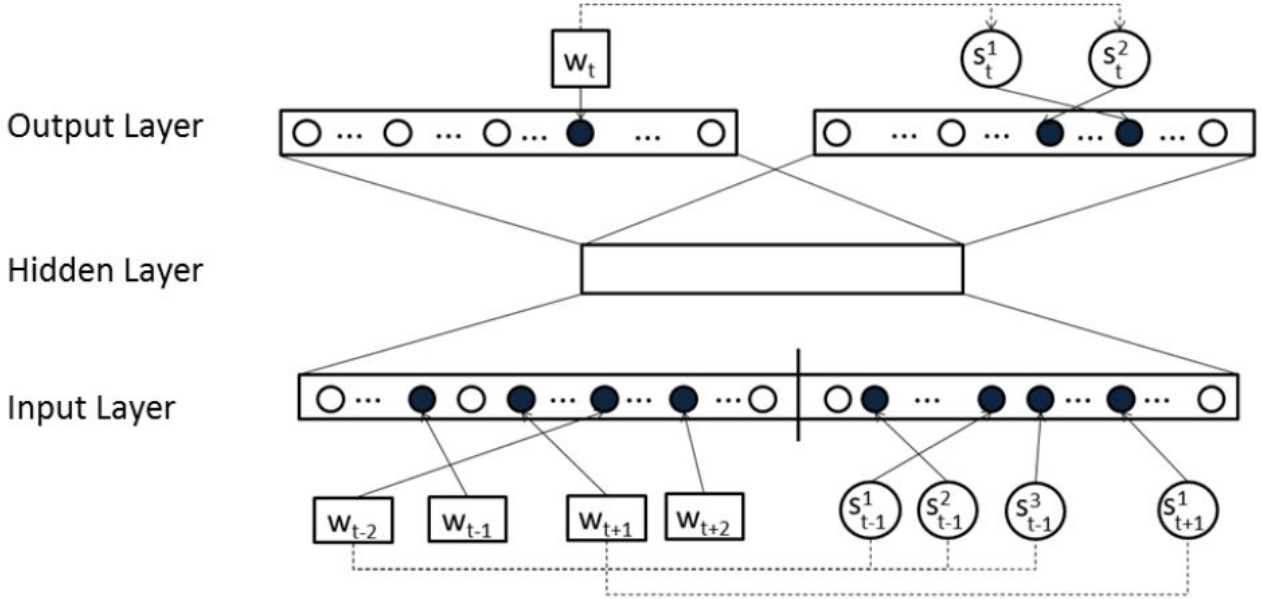
Loss function:

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

Table of Contents

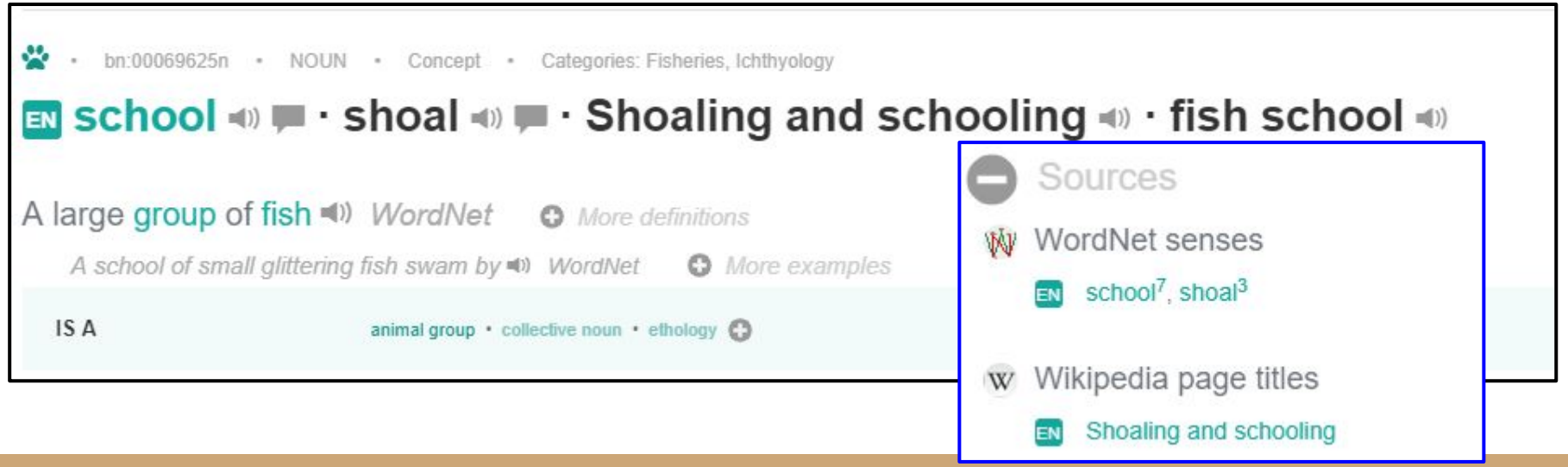
Experimental settings

- Determining ideal settings for evaluation



Experimental settings

- Determining ideal settings for training / testing
  - CBOW: dimensionality = 300 , window size = 8, hierarchical softmax
  - Corpus: UMBC WebBase corpus of 3B-words with English paragraphs
  - Semantic network: BabelNet 3.0, integrating Wikipedia and WordNet



Experimental settings

- Determining ideal settings for training / testing
  - CBOW: dimensionality = 300 , window size = 8, hierarchical softmax
  - Corpus: UMBC WebBase corpus of 300M-words with English paragraphs
  - Semantic network: BabelNet 3.0, integrating Wikipedia and WordNet
  - Training task: word similarity:
    - WordSim-353
    - RG-65

    } Contain word pairs manually annotated with similarity scores

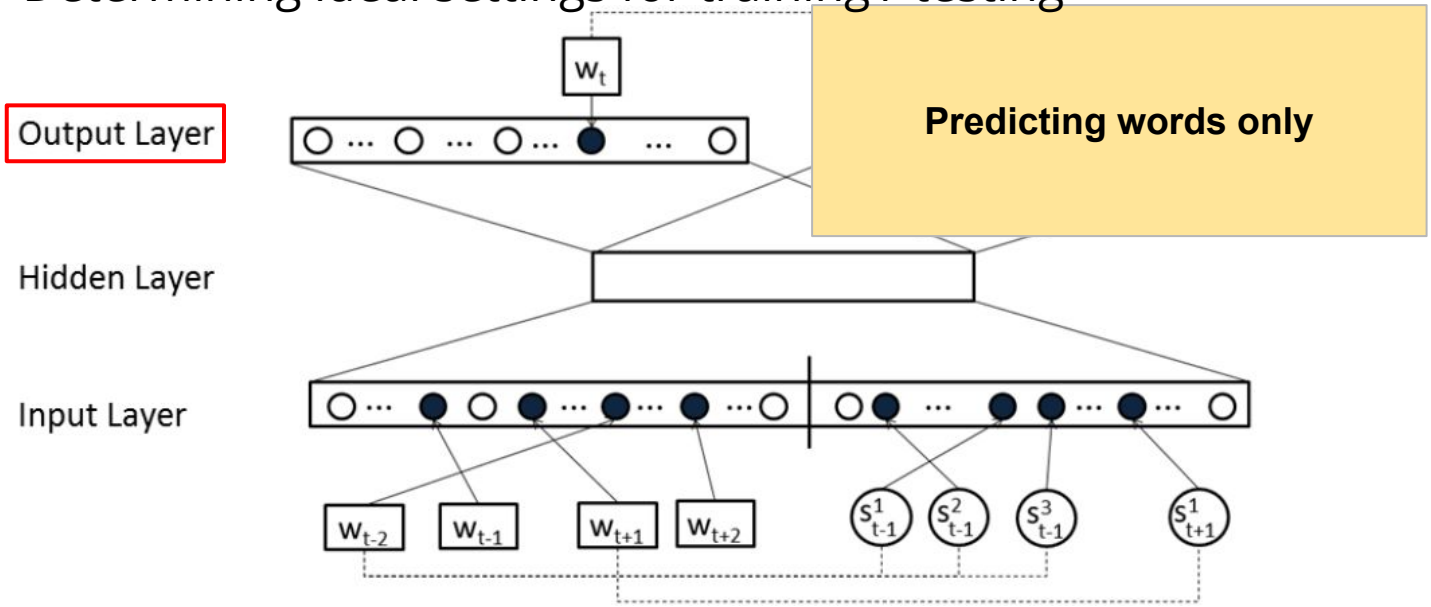
  - Closest senses strategy for similarity scores:

$$sim(w_1, w_2) = max_{s_1 \in S_{w_1}, s_2 \in S_{w_2}} cos(\overrightarrow{s_1}, \overrightarrow{s_2})$$

=> distance between two words is the maximum similarity between two of their senses

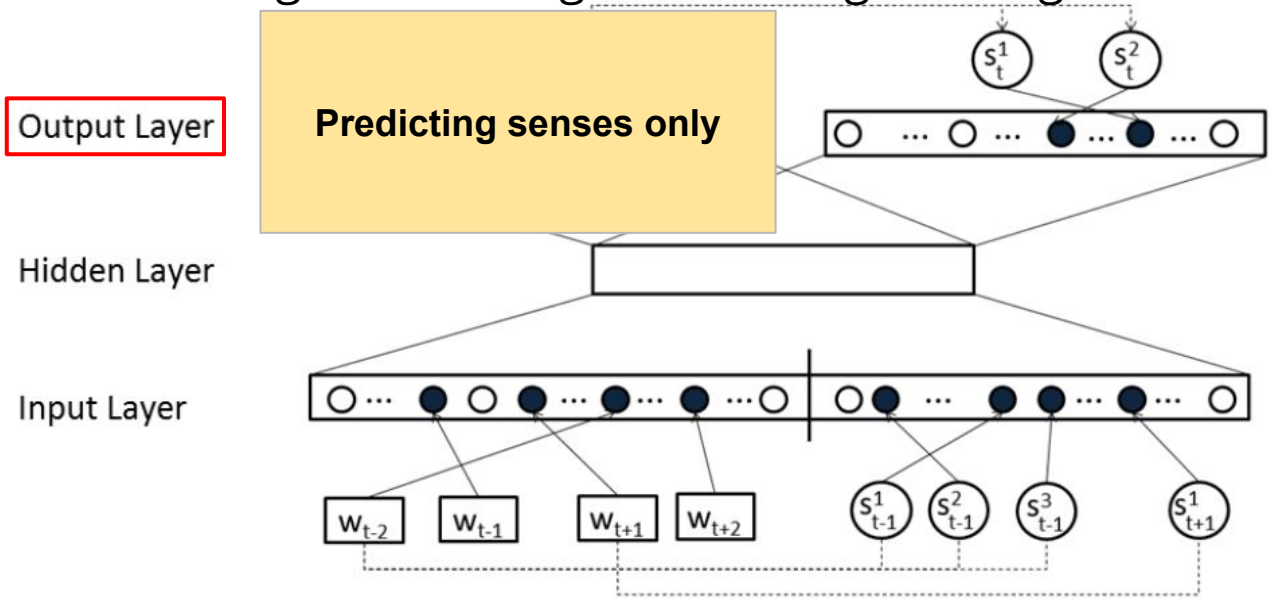
Experimental settings

- Determining ideal settings for training / testing



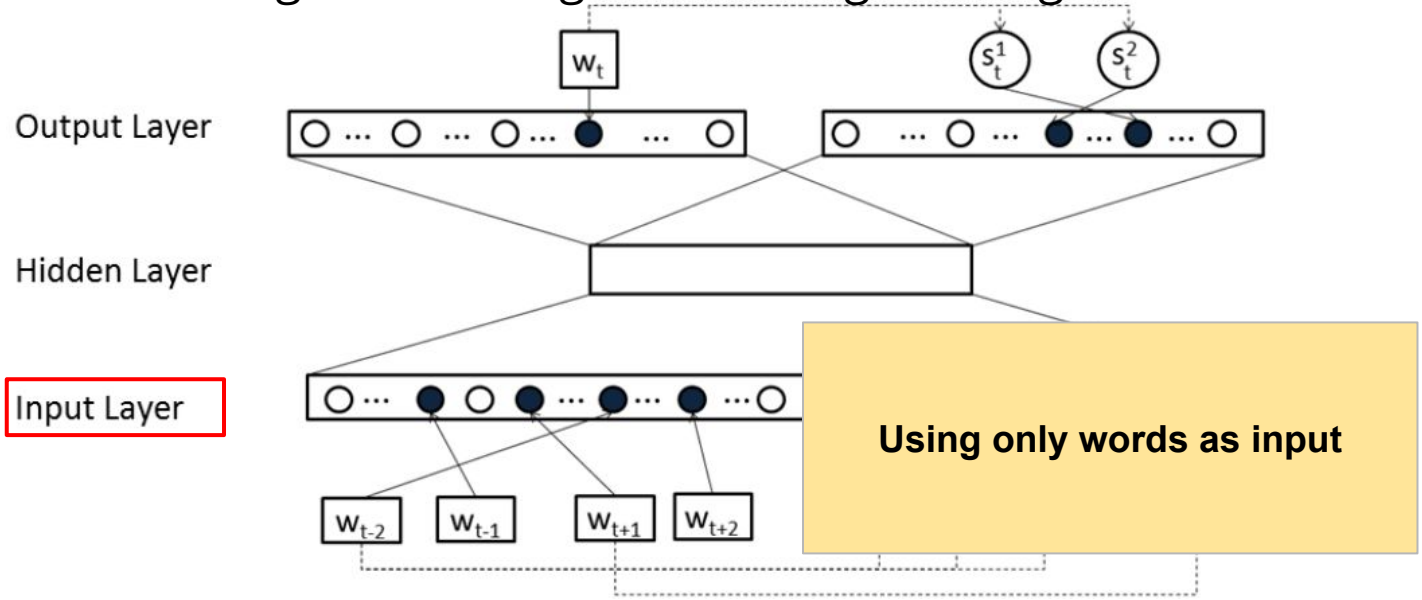
Experimental settings

- Determining ideal settings for training / testing



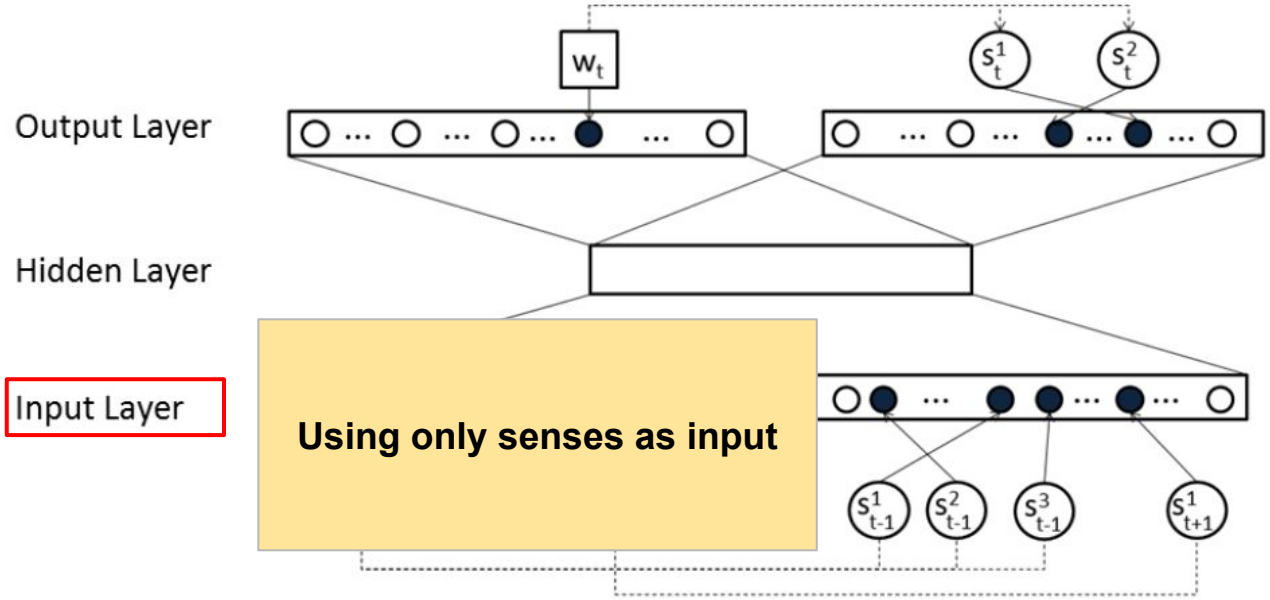
Experimental settings

- Determining ideal settings for training / testing



Experimental settings

- Determining ideal settings for training / testing



Experimental settings

- Determining ideal settings for training / testing

		Output											
		Words				Senses				Both			
		WS-Sim		RG-65		WS-Sim		RG-65		WS-Sim		RG-65	
		r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
Input	**Words**	0.49	0.48	0.65	0.66	0.56	0.56	0.67	0.67	0.54	0.53	0.66	0.65
	Senses	0.69	0.69	0.70	0.71	0.69	0.70	0.70	**0.74**	**0.72**	**0.71**	**0.71**	**0.74**
	Both	0.60	0.65	0.67	0.70	0.62	0.65	0.66	0.67	0.65	**0.71**	0.68	0.70

Table 1: Pearson (r) and Spearman (ρ) correlation performance of the nine configurations of SW2V

Experimental settings

- Determining ideal settings for training / testing

		Output											
		Words				Senses				Both			
		WS-Sim		RG-65		WS-Sim		RG-65		WS-Sim		RG-65	
		r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
Input	**Words**	0.49	0.48	0.65	0.66	0.56	0.56	0.67	0.67	0.54	0.53	0.66	0.65
	Senses	0.69	0.69	0.70	0.71	0.69	0.70	0.70	**0.74**	**0.72**	**0.71**	**0.71**	**0.74**
	Both	0.60	0.65	0.67	0.70	0.62	0.65	0.66	0.67	0.65	**0.71**	0.68	0.70

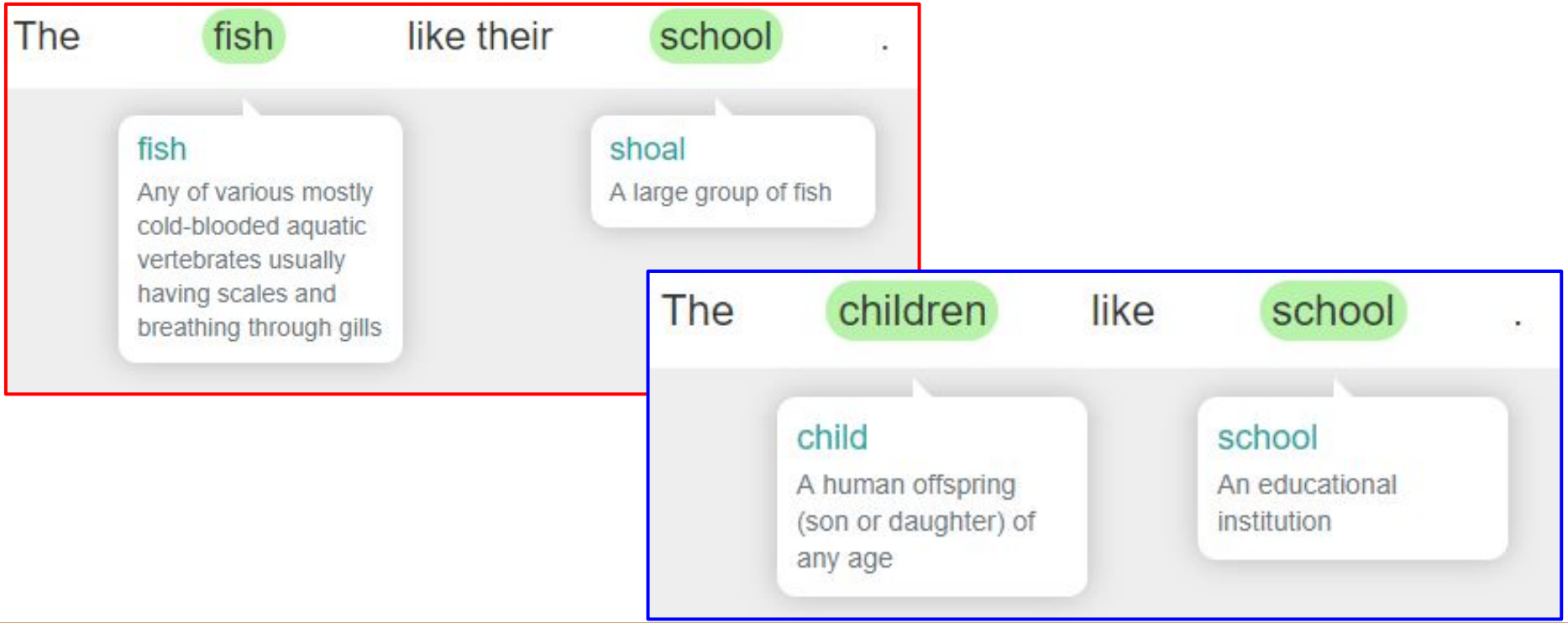
Table 1: Pearson (r) and Spearman (ρ) correlation performance of the nine configurations of SW2V

Experimental settings

- Determining ideal settings for training / testing
- Testing the impact of the *shallow word-sense connectivity algorithm*
  - Comparison to already disambiguated texts
    - **Babelfy:** State-of-the-art disambiguation / entity linking system

Experimental Settings

- Babelfy:



Experimental settings

	WS-Sim		RG-65	
	r	ρ	r	ρ
Shallow	**0.72**	**0.71**	**0.71**	**0.74**
Babelfy	0.65	0.63	0.69	0.70
Babelfy*	0.63	0.61	0.65	0.64

Table 2: Pearson (r) and Spearman (ρ) correlation performance of SW2V integrating our *shallow* word-sense connectivity algorithm (default), Babelfy, or Babelfy*.

Babelfy = default version using *Most Common Sense* heuristic as back-off strategy

Babelfy* = includes only instances that exceed a confidence threshold

Experimental settings

	WS-Sim		RG-65	
	r	ρ	r	ρ
Shallow	**0.72**	**0.71**	**0.71**	**0.74**
Babelfy	0.65	0.63	0.69	0.70
Babelfy*	0.63	0.61	0.65	0.64

Table 2: Pearson (r) and Spearman (ρ) correlation performance of SW2V integrating our *shallow* word-sense connectivity algorithm (default), Babelfy, or Babelfy*.

Babelfy = default version using *Most Common Sense* heuristic as back-off strategy

Babelfy* = includes only instances that exceed a confidence threshold

Experimental settings

- Determining ideal settings for training / testing
- Testing the impact of the *shallow word-sense connectivity algorithm*
  - Comparison to already disambiguated texts
    - Babelfy: State-of-the-art disambiguation / entity linking system
  - **Provides an improvement of accuracy**
  - **Improvement of speed: takes only tenth of the time**

Table of Contents

Evaluation

- Three different tasks:

    A. Word Similarity Task
    B. Sense Clustering
    C. Word and sense interconnectivity

Word Similarity Task

- Evaluate sense representations on the standard SimLex-999 and MEN word similarity datasets.
  - Both datasets contain 999 and 3000 word pairs, respectively
  - The two largest similarity datasets
- Closest sense strategy
- Cosine similarity
- Retrofitted version of word2vec word vectors
  - WordNet and BabelNet

Word Similarity Task

	System	Corpus	SimLex-999 r	SimLex-999 ρ	MEN r	MEN ρ
Senses	$SW2V_{BN}$	UMBC	**0.49**	**0.47**	0.75	0.75
	$SW2V_{WN}$	UMBC	0.46	0.45	**0.76**	**0.76**
	AutoExtend	UMBC	0.47	0.45	0.74	0.75
	AutoExtend	Google-News	0.46	0.46	0.68	0.70
	$SW2V_{BN}$	Wikipedia	0.47	0.43	0.71	0.73
	$SW2V_{WN}$	Wikipedia	0.47	0.43	0.71	0.72
	SensEmbed	Wikipedia	0.43	0.39	0.65	0.70
	Chen et al. (2014)	Wikipedia	0.46	0.43	0.62	0.62
Words	Word2Vec	UMBC	0.39	0.39	0.75	0.75
	$Retrofitting_{BN}$	UMBC	0.47	0.46	0.75	**0.76**
	$Retrofitting_{WN}$	UMBC	0.47	0.46	**0.76**	**0.76**
	Word2Vec	Wikipedia	0.39	0.38	0.71	0.72
	$Retrofitting_{BN}$	Wikipedia	0.35	0.32	0.66	0.66
	$Retrofitting_{WN}$	Wikipedia	0.47	0.44	0.73	0.73

Sense Clustering

- High level of granularity in current lexical resources
- Idea: clustering senses to possibly improve on downstream tasks
- Wikipedia sense clustering task: 500-pair & SemEval datasets
  - Binary classification : two Wikipedia pages
    - Clustering them into a single instance
    - Not clustering them
  - Similarity between synset embeddings has to exceed given threshold tuned with 500-pair dataset
  - Evaluation on SemEval clustering test set consisting of 925 pairs of highly ambiguous words

Sense Clustering

$$\frac{2 \cdot precision \cdot recall}{precision + recall}$$

	Accuracy	F-Measure
SW2V	**87.8**	**63.9**
SensEmbed	82.7	40.3
NASARI	87.0	62.5
Multi-SVM	85.5	-
Mono-SVM	83.5	-
Baseline	17.5	29.8

Baseline = every pair clustered together

Pre-trained state-of-the-art sense embeddings, also including Babelnet

Supervised Support Vector Machine classifier trained on labeled mono- or multilingual Wikipedia data

Table 4: Accuracy and F-Measure percentages of different systems on the SemEval Wikipedia sense clustering dataset.

➔ **Semantics of word senses are captured well on this task**

Word and sense interconnectivity

- Previously, no research on **semantic coherence** of vector space obtained by jointly learning word and sense embeddings
- Word Sense Disambiguation task
- *Most Common Sense* (MCS) baseline obtained from our shared vector space
  - MCS of a word closer to its vector than any other sense
  - Formally: $MCS(w) = argmax_{s \in S_w} cos(\vec{w}, \vec{s})$
  - Integrated into various NLP applications, e.g. Word Sense Disambiguation, Entity Linking
  - Automatic identification from non-annotated text useful for low resource languages or large datasets

Word and sense interconnectivity

	SemEval-07	SemEval-13
SW2V	**39.9**	**54.0**
AutoExtend	17.6	31.0
Baseline	24.8	34.9

Table 5: F-Measure percentage of different MCS strategies on the SemEval-2007 and SemEval-2013 WSD datasets.

Baseline = randomly assigning sense as Most Common Sense

Pre-trained state-of-the-art sense embeddings, also including Babelnet

➔ SW2V seems to be a good architecture to capture *Most Common Senses*

Word and sense interconnectivity

Qualitative comparison to AutoExtend:

company^2_n *(military unit)*		*school*^7_n *(group of fish)*	
AutoExtend	**SW2V**	**AutoExtend**	**SW2V**
company^9_n	battalion^1_n	school	schools^7_n
company	battalion	school^4_n	sharks^1_n
company^8_n	regiment^1_n	school^6_n	sharks
company^6_n	detachment^4_n	school^1_v	shoals^3_n
company^7_n	platoon^1_n	school^3_n	fish^1_n
company^1_v	brigade^1_n	elementary	dolphins^1_n
firm	regiment	schools	pods^3_n
business^1_n	corps^1_n	elementary^3_a	eels
firm^2_n	brigade	school^5_n	dolphins
company^1_n	platoon	elementary^1_a	whales^2_n

Table 6: Ten closest word and sense embeddings to the senses *company*^2_n (military unit) and *school*^7_n (group of fish).

Word and sense interconnectivity

Qualitative comparison to AutoExtend:

AutoExtend creates clusters that include all senses of a word together, including respectively related concepts

$company_n^2$ (military unit)		$school_n^7$ (group of fish)	
AutoExtend	**SW2V**	**AutoExtend**	**SW2V**
$company_n^9$	$battalion_n^1$	school	$schools_n^7$
company	battalion	$school_n^4$	$sharks_n^1$
$company_n^8$	$regiment_n^1$	$school_n^6$	sharks
$company_n^6$	$detachment_n^4$	$school_v^1$	$shoals_n^3$
$company_n^7$	$platoon_n^1$	$school_n^3$	$fish_n^1$
$company_v^1$	$brigade_n^1$	elementary	$dolphins_n^1$
firm	regiment	schools	$pods_n^3$
$business_n^1$	$corps_n^1$	$elementary_a^3$	eels
$firm_n^2$	brigade	$school_n^5$	dolphins
$company_n^1$	platoon	$elementary_a^1$	$whales_n^2$

Table 6: Ten closest word and sense embeddings to the senses $company_n^2$ (military unit) and $school_n^7$ (group of fish).

Word and sense interconnectivity

Qualitative comparison to AutoExtend:

AutoExtend creates clusters that include **all senses** of a word together, including respectively related concepts

Semantic cluster fitting to the **specific** sense of the word

company^2_n *(military unit)*		*school*^7_n *(group of fish)*	
AutoExtend	**SW2V**	**AutoExtend**	**SW2V**
company^9_n	battalion^1_n	school	schools^7_n
company	battalion	school^4_n	sharks^1_n
company^8_n	regiment^1_n	school^6_n	sharks
company^6_n	detachment^4_n	school^1_v	shoals^3_n
company^7_n	platoon^1_n	school^3_n	fish^1_n
company^1_v	brigade^1_n	elementary	dolphins^1_n
firm	regiment	schools	pods^3_n
business^1_n	corps^1_n	elementary^3_a	eels
firm^2_n	brigade	school^5_n	dolphins
company^1_n	platoon	elementary^1_a	whales^2_n

Table 6: Ten closest word and sense embeddings to the senses *company*^2_n (military unit) and *school*^7_n (group of fish).

Table of Contents

Conclusion

- Senses and Words to Vectors
  - a neural model which uses both text corpora and knowledge from semantic networks to learn vectors representations for words and senses via joint training
- Architecture is scalable to also use larger semantic networks, such as BabelNet
- No annotated data is needed
- Good performance on capturing similarities between both words and senses as well as their interconnectivity

Conclusion

- Opinion:
  - Good architecture, but they only use WordNet and BabelNet for training, and WordNet and BabelNet work really well together
  - They say it is flexible, but they don’t actually prove it
    - Good for resource-poor languages, but they don’t show any experiments proving it
  - They test it with CBOW, how about Skip-gram? Both models perform differently with different tasks
  - Antonyms get very high similarity scores
  - There is a way to force low similarity scores

References

- Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci and Roberto Navigli. (2017). Embedding Words and Senses Together via Joint Knowledge-Enhanced Training.*Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Pages 100–111
- https://en.wikipedia.org/wiki/F1_score (03.07.2019)
- https://babelnet.org/ (03.07.2019)
- http://babelfy.org/ (03.07.2019)