Finding Word Sense Embeddings of Known Meaning

A method for refitting word sense embeddings, using a single example, by application of Bayes' theorem to the language model

Lyndon White,
Roberto Togneri, Wei Liu, Mohammed Bennamoun

School of Electical, Electronic and Computer Engineering
The University of Western Australia

Words don’t only have one meaning

Kid *(Noun)*

1. (a young person of either sex) "she writes books for children"; "they’re just kids"; "‘tiddler’ is a British term for youngster"
2. (English dramatist (1558-1594))
3. (a human offspring (son or daughter) of any age) "they had three children"; "they were able to send their kids to college"
4. (young goat)

George A Miller. “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

Word embeddings represent each word as a single vector

SkipGram Language Model:

- Input a word w_T
- Output probabilities of words appearing in its context $P(w_i \mid w_T)$

$$\overbrace{\overbrace{\texttt{wow}}^{w_1}\ \overbrace{\texttt{the}}^{w_2}\ \overbrace{\texttt{wool}}^{w_3}\ \overbrace{\texttt{from}}^{w_4}\ \overbrace{\texttt{the}}^{w_5}\quad \underbrace{\texttt{kid}}_{\textit{target word}}\quad \overbrace{\texttt{is}}^{w_6}\ \overbrace{\texttt{so}}^{w_7}\ \overbrace{\texttt{soft}}^{w_8}\ \overbrace{\texttt{and}}^{w_9}\ \overbrace{\texttt{fluffy}}^{w_{10}}}^{\mathbf{c}}$$

Word embeddings represent each word as a single vector

SkipGram Language Model:

- Input a word w_T
- Output probabilities of words appearing in its context $P(w_i \mid w_T)$

Word Embeddings Implementation

- Represent each input word as a vector
- Train a neural network to estimate $P(w_i \mid w_T)$
- Back-prop finds values for the input vector – i.e. good representation for the word

$$\overbrace{\overbrace{\texttt{wow}}^{w_1}\ \overbrace{\texttt{the}}^{w_2}\ \overbrace{\texttt{wool}}^{w_3}\ \overbrace{\texttt{from}}^{w_4}\ \overbrace{\texttt{the}}^{w_5}\quad \underbrace{\texttt{kid}}_{\textit{target word}}\quad \overbrace{\texttt{is}}^{w_6}\ \overbrace{\texttt{so}}^{w_7}\ \overbrace{\texttt{soft}}^{w_8}\ \overbrace{\texttt{and}}^{w_9}\ \overbrace{\texttt{fluffy}}^{w_{10}}}^{\mathbf{c}}$$

Word sense embeddings represent each word as a multiple vectors

- Each word has multiple senses $\{u_1, u_2, \dots, u_n\}$
- SkipGram Language Model becomes
  - Input a word sense u_i
  - Output probabilities of words appearing in its context $P(w_i \mid u_j)$

$$\overbrace{\overbrace{\texttt{wow}}^{w_1}\ \overbrace{\texttt{the}}^{w_2}\ \overbrace{\texttt{wool}}^{w_3}\ \overbrace{\texttt{from}}^{w_4}\ \overbrace{\texttt{the}}^{w_5}\quad \underbrace{\texttt{kid}}_{\textit{target word}}\quad \overbrace{\texttt{is}}^{w_6}\ \overbrace{\texttt{so}}^{w_7}\ \overbrace{\texttt{soft}}^{w_8}\ \overbrace{\texttt{and}}^{w_9}\ \overbrace{\texttt{fluffy}}^{w_{10}}}^{\mathbf{c}}$$

Sergey Bartunov et al. "Breaking Sticks and Ambiguities with Adaptive Skip-gram". In: *CoRR* abs/1502.07257 (2015).

Many sense embeddings don’t produce human recognisable senses

- Embeddings are learnt by modelling what words occur near the sense

Many sense embeddings don't produce human recognisable senses

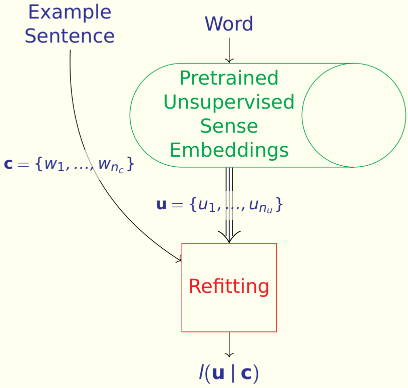
- Embeddings are learnt by modelling what words occur near the sense
- No control over the meanings of the senses
  - Cover overlapping definitions
  - Find overly narrow meanings
  - Capture rare jargon uses

Many sense embeddings don't produce human recognisable senses

- Embeddings are learnt by modelling what words occur near the sense
- No control over the meanings of the senses
  - Cover overlapping definitions
  - Find overly narrow meanings
  - Capture rare jargon uses
- Useful, but not interoperable with lexical knowledge bases.

We will solve this by *refitting* them to be for the sense we mean

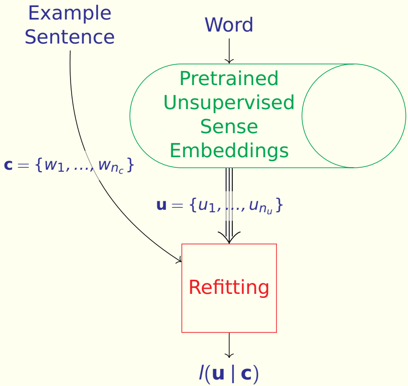
- Refitting constructs new sense embeddings out of the old.
- It uses the probabilities of example sentence occuring.
- The new embedding aligns to the meaning in that sentence.



Refitting uses a probability weighted sum

New sense embedding:

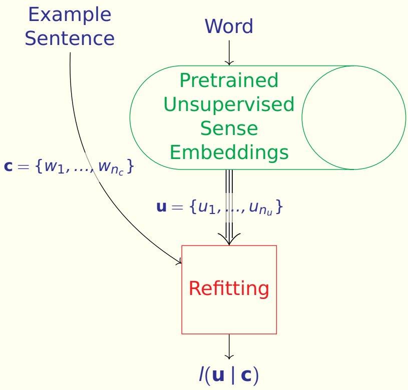
$$l(\mathbf{u} \mid \mathbf{c}) = \sum_{\forall u_i \in \mathbf{u}} u_i P(u_i \mid \mathbf{c})$$



$$\overbrace{\overbrace{\texttt{wow}}^{w_1}\ \overbrace{\texttt{the}}^{w_2}\ \overbrace{\texttt{wool}}^{w_3}\ \overbrace{\texttt{from}}^{w_4}\ \overbrace{\texttt{the}}^{w_5}\quad \underbrace{\texttt{kid}}_{\textit{target word}}\quad \overbrace{\texttt{is}}^{w_6}\ \overbrace{\texttt{so}}^{w_7}\ \overbrace{\texttt{soft}}^{w_8}\ \overbrace{\texttt{and}}^{w_9}\ \overbrace{\texttt{fluffy}}^{w_{10}}}^{\mathbf{c}}$$

The probabilities are found using Bayes' theorem

Language model: $P(w_i \mid u_i)$



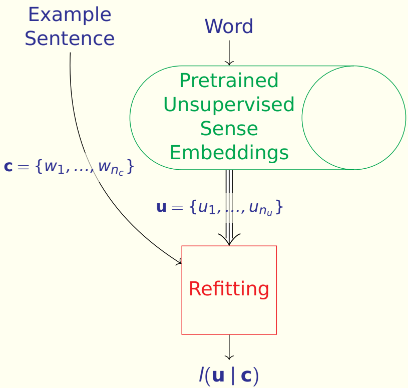
$$\overbrace{\underbrace{\texttt{wow}}^{w_1}\ \underbrace{\texttt{the}}^{w_2}\ \underbrace{\texttt{wool}}^{w_3}\ \underbrace{\texttt{from}}^{w_4}\ \underbrace{\texttt{the}}^{w_5}\ \underbrace{\texttt{kid}}_{\textit{target word}}\ \underbrace{\texttt{is}}^{w_6}\ \underbrace{\texttt{so}}^{w_7}\ \underbrace{\texttt{soft}}^{w_8}\ \underbrace{\texttt{and}}^{w_9}\ \underbrace{\texttt{fluffy}}^{w_{10}}}^{\mathbf{c}}$$

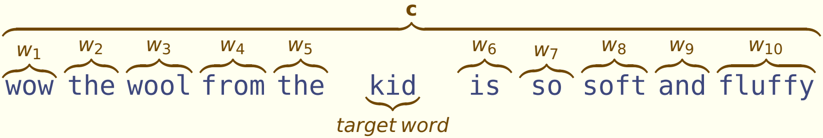
The probabilities are found using Bayes' theorem

Language model: $P(w_i \mid u_i)$

Conditional Independence:

$$P(\mathbf{c} \mid u_i) = \prod_{\forall w_j \in \mathbf{c}} P(w_j \mid u_i)$$





The probabilities are found using Bayes' theorem

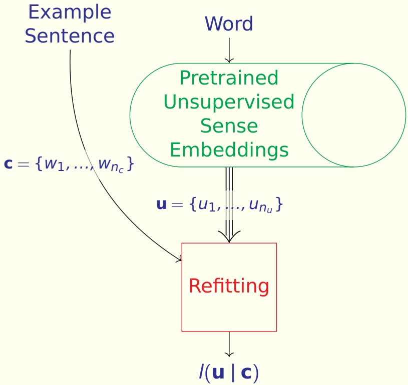
Language model: $P(w_i \mid u_i)$

Conditional Independence:

$$P(\mathbf{c} \mid u_i) = \prod_{\forall w_j \in \mathbf{c}} P(w_j \mid u_i)$$

Bayes Theorem:

$$P(u_i \mid \mathbf{c}) = \frac{P(\mathbf{c} \mid u_i) P(u_i)}{\sum_{u_j \in \mathbf{s}} P(\mathbf{c} \mid u_j) P(u_j)}$$



$$\overbrace{\overbrace{\texttt{wow}}^{w_1}\ \overbrace{\texttt{the}}^{w_2}\ \overbrace{\texttt{wool}}^{w_3}\ \overbrace{\texttt{from}}^{w_4}\ \overbrace{\texttt{the}}^{w_5}\ \underbrace{\texttt{kid}}_{\textit{target word}}\ \overbrace{\texttt{is}}^{w_6}\ \overbrace{\texttt{so}}^{w_7}\ \overbrace{\texttt{soft}}^{w_8}\ \overbrace{\texttt{and}}^{w_9}\ \overbrace{\texttt{fluffy}}^{w_{10}}}^{\mathbf{c}}$$

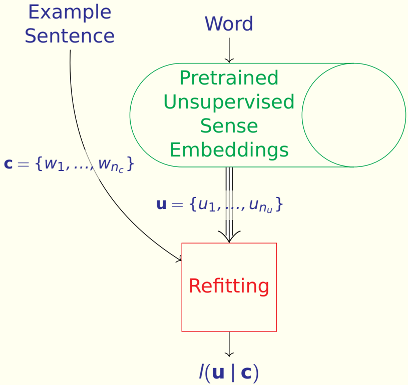
The probabilities are found using Bayes' theorem

Bayes Theorem:

$$P(u_i \mid \mathbf{c}) = \frac{P(\mathbf{c} \mid u_i) P(u_i)}{\sum_{u_j \in \mathbf{s}} P(\mathbf{c} \mid u_j) P(u_j)}$$

Refitted Sense Embedding:

$$l(\mathbf{u} \mid \mathbf{c}) = \sum_{\forall u_i \in \mathbf{u}} u_i P(u_i \mid \mathbf{c})$$



$$\overbrace{\overbrace{\texttt{wow}}^{w_1}\ \overbrace{\texttt{the}}^{w_2}\ \overbrace{\texttt{wool}}^{w_3}\ \overbrace{\texttt{from}}^{w_4}\ \overbrace{\texttt{the}}^{w_5}\quad \underbrace{\texttt{kid}}_{\textit{target word}}\quad \overbrace{\texttt{is}}^{w_6}\ \overbrace{\texttt{so}}^{w_7}\ \overbrace{\texttt{soft}}^{w_8}\ \overbrace{\texttt{and}}^{w_9}\ \overbrace{\texttt{fluffy}}^{w_{10}}}^{\mathbf{c}}$$

The posterior distribution (over senses) is too sharp, so we smooth it

Original:

Context Likelihood:

$$P(\mathbf{c} \mid u_i) = \prod_{\forall w_j \in \mathbf{c}} P(w_j \mid u_i)$$

Sense Likelihood:

$$P(u_i \mid \mathbf{c}) = \frac{P(\mathbf{c} \mid u_i) P(u_i)}{\sum_{u_j \in \mathbf{u}} P(\mathbf{c} \mid u_j) P(u_j)}$$

The posterior distribution (over senses) is too sharp, so we smooth it

Original:

Context Likelihood:

$$P(\mathbf{c} \mid u_i) = \prod_{\forall w_j \in \mathbf{c}} P(w_j \mid u_i)$$

Sense Likelihood:

$$P(u_i \mid \mathbf{c}) = \frac{P(\mathbf{c} \mid u_i) P(u_i)}{\sum_{u_j \in \mathbf{u}} P(\mathbf{c} \mid u_j) P(u_j)}$$

Smoothed:

Context Likelihood:

$$P_S(\mathbf{c} \mid u_i) = \prod_{\forall w_j \in \mathbf{c}} \sqrt[|\mathbf{c}|]{P(w_j \mid u_i)}$$

The posterior distribution (over senses) is too sharp, so we smooth it

Original:

Context Likelihood:

$$P(\mathbf{c} \mid u_i) = \prod_{\forall w_j \in \mathbf{c}} P(w_j \mid u_i)$$

Sense Likelihood:

$$P(u_i \mid \mathbf{c}) = \frac{P(\mathbf{c} \mid u_i) P(u_i)}{\sum_{u_j \in \mathbf{u}} P(\mathbf{c} \mid u_j) P(u_j)}$$

Smoothed:

Context Likelihood:

$$P_S(\mathbf{c} \mid u_i) = \prod_{\forall w_j \in \mathbf{c}} \sqrt[|\mathbf{c}|]{P(w_j \mid u_i)}$$

Sense Likelihood:

$$P_S(s_i \mid \mathbf{c}) = \frac{\sqrt[|\mathbf{c}|]{P(\mathbf{c} \mid u_i) P(u_i)}}{\sum_{u_j \in \mathbf{u}} \sqrt[|\mathbf{c}|]{P(\mathbf{c} \mid u_j) P(u_j)}}$$

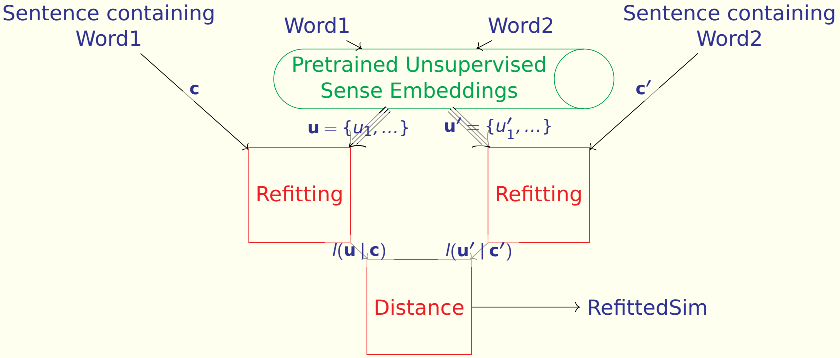
Similarity with context, is the task of ranking how similar a word is, given its usage

$$\overbrace{\overbrace{\texttt{stone}}^{w_1}\ \overbrace{\texttt{axes}}^{w_w}\ \overbrace{\texttt{was}}^{w_3}\ \overbrace{\texttt{a}}^{w_4}\ \overbrace{\texttt{major}}^{w_5}\ \underbrace{\texttt{advance}}_{word\,1}\ \overbrace{\texttt{because}}^{w_6}\ \overbrace{\texttt{it}}^{w_7}\ \overbrace{\texttt{allowed}}^{w_8}\ \overbrace{\texttt{forest}}^{w_9}\ \overbrace{\texttt{clearance}}^{w_{10}}}^{\mathbf{c}}$$

$$\overbrace{\overbrace{\texttt{forces}}^{w'_1}\ \overbrace{\texttt{could}}^{w'_2}\ \overbrace{\texttt{make}}^{w'_3}\ \overbrace{\texttt{no}}^{w'_4}\ \overbrace{\texttt{further}}^{w'_5}\ \underbrace{\texttt{headway}}_{word\,2}\ \overbrace{\texttt{without}}^{w'_6}\ \overbrace{\texttt{resting}}^{w'_7}\ \overbrace{\texttt{and}}^{w'_8}\ \overbrace{\texttt{regrouping}}^{w'_9}\ \overbrace{\texttt{.}}^{w'_{10}}}^{\mathbf{c}'}$$

Huang et al. 2012.

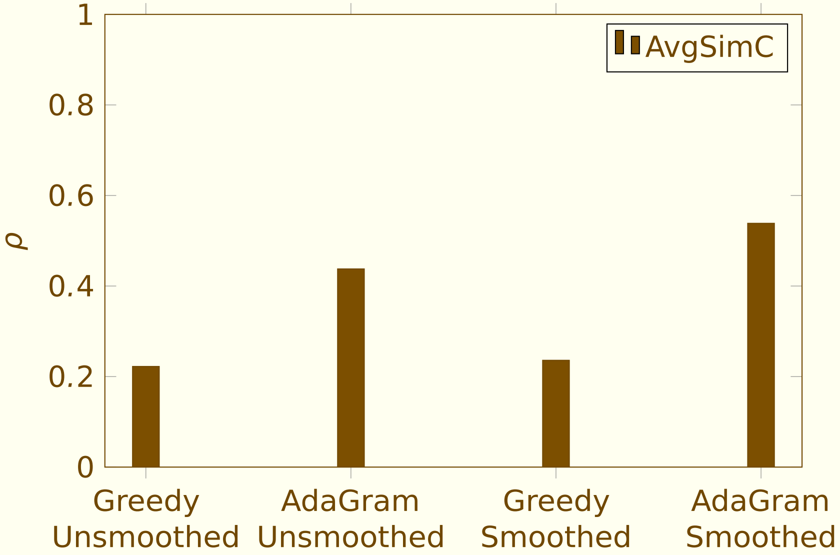
Use for word similarity with context



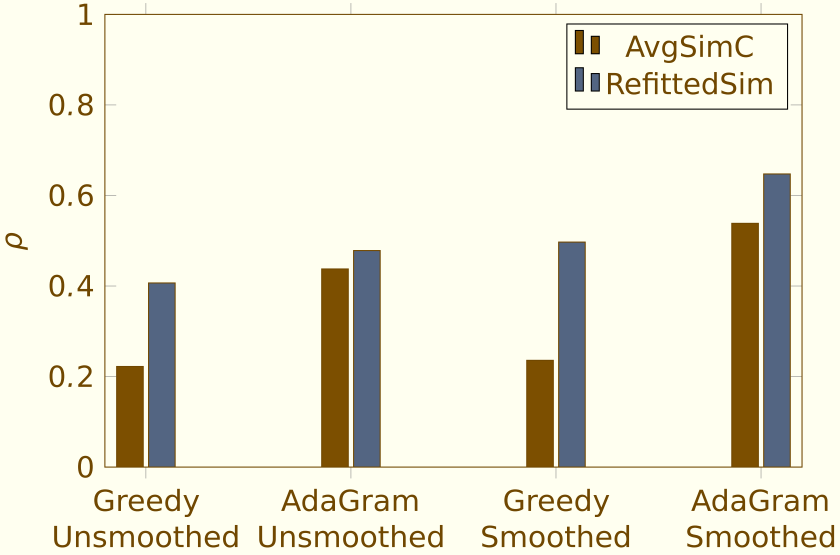
$$\mathrm{RefittedSim}((\mathbf{u},\mathbf{c}),(\mathbf{u'},\mathbf{c'})) = d(l(\mathbf{u}\mid\mathbf{c}), l(\mathbf{u'}\mid\mathbf{c'}))$$

$$\mathrm{RefittedSim}((\mathbf{u},\mathbf{c}),(\mathbf{u'},\mathbf{c'})) = d\left(\sum_{u_i\in\mathbf{u}} u_i P(u_i\mid\mathbf{c}),\ \sum_{u'_j\in\mathbf{u'}} u_i P(u'_j\mid\mathbf{c'})\right)$$

Results on word similarity with context



Results on word similarity with context



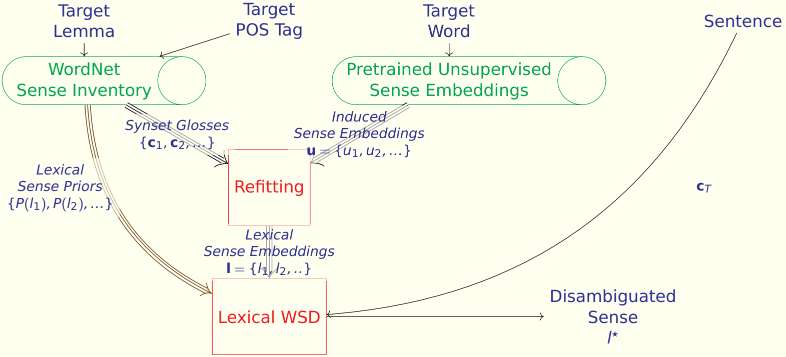
WSD is the task of determining which sense is being used

Kid *(Noun)*

1. (a young person of either sex) "she writes books for children"; "they’re just kids"; "‘tiddler’ is a British term for youngster"
2. (English dramatist (1558-1594))
3. (a human offspring (son or daughter) of any age) "they had three children"; "they were able to send their kids to college"
4. (young goat)

$$\overbrace{\overbrace{\texttt{wow}}^{w_1}\ \overbrace{\texttt{the}}^{w_2}\ \overbrace{\texttt{wool}}^{w_3}\ \overbrace{\texttt{from}}^{w_4}\ \overbrace{\texttt{the}}^{w_5}\quad \underbrace{\texttt{kid}}_{\textit{target word}}\quad \overbrace{\texttt{is}}^{w_6}\ \overbrace{\texttt{so}}^{w_7}\ \overbrace{\texttt{soft}}^{w_8}\ \overbrace{\texttt{and}}^{w_9}\ \overbrace{\texttt{fluffy}}^{w_{10}}}^{\mathbf{c}}$$

Use of refitted senses for word sense disambiguation

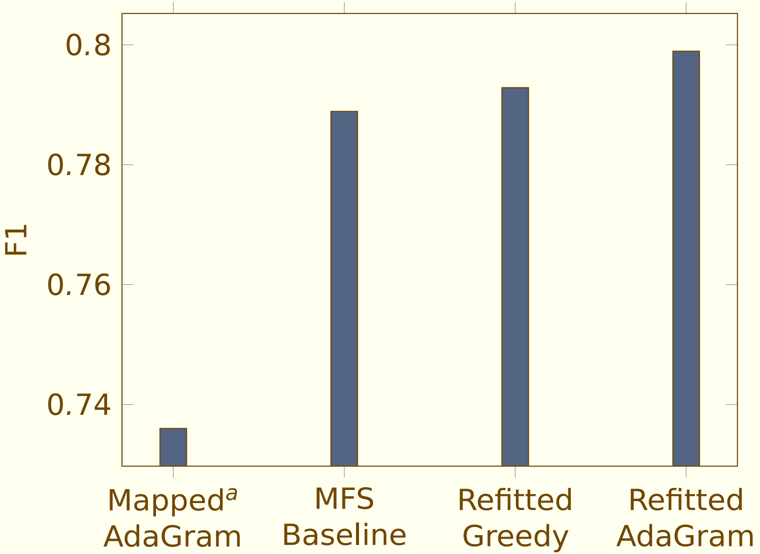


$$l^\star(\mathbf{l}, \mathbf{c}_T) = \operatorname*{arg\,max}_{\forall l_i \in \mathbf{l}} P(l_i | \mathbf{c}_T)$$

$$l^\star(\mathbf{l}, \mathbf{c}_T) = \operatorname*{arg\,max}_{\forall l_i \in \mathbf{l}} \frac{P(\mathbf{c}_T \mid l_i) P(l_i)}{\sum_{\forall l_j \in \mathbf{l}} P(\mathbf{c}_T \mid l_j) P(l_j)}$$

Results for word sense disambiguation

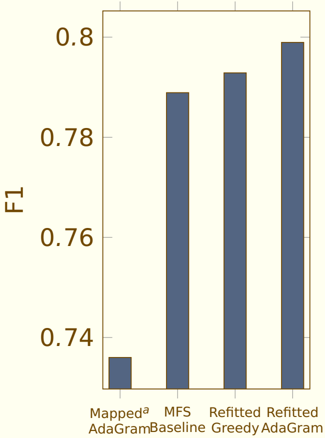
SemEval 2007 Task 7



[a] Agirre et al. 2006

Discussion of the WSD results

- Results are not great: an improvement of a few percent over the baseline.
- With that said, this is an almost unsupervised method.
- The *geometric smoothing* to an extent trades-off between he prior (which is linked to MFS).



[a] Agirre et al. 2006

Conclusion

- RefittedSim, faster and higher correlation with human judgement than AvgSimC.
- WSD results using refitted not competitive with supervised methods.
- This problem of aligning induced senses to lexical senses is important.

Appendix

RefittedSim vs AvgSimC

RefittedSim

$$\mathrm{RefittedSim}((\mathbf{u},\mathbf{c}),(\mathbf{u}',\mathbf{c}')) = d\left(\sum_{u_i\in\mathbf{u}} u_i P(u_i \mid \mathbf{c}),\ \sum_{u'_j\in\mathbf{u}'} u_i P(u'_j \mid \mathbf{c}')\right)$$

Time Complexity: $O(n\,\|\mathbf{c}\| + n'\,\|\mathbf{c}'\|)$

AvgSimC

$$\mathrm{AvgSimC}((\mathbf{u},\mathbf{c}),(\mathbf{u}',\mathbf{c}')) = \frac{1}{n\times n'}\sum_{u_i\in\mathbf{u}}\sum_{u'_j\in\mathbf{u}'} P(u_i \mid \mathbf{c})\,P(u'_j \mid \mathbf{c}')\,d(u_i,u'_j)$$

Time Complexity: $O(n\,\|\mathbf{c}\| + n'\,\|\mathbf{c}'\| + n\times n')$

Results on word similarity with context

Method	Geometric Smoothing	Use Prior	AvgSimC	RefittedSim
AdaGram	T	T	**53.8**	64.8
AdaGram	T	F	36.1	**65.0**
AdaGram	F	T	43.8	47.8
AdaGram	F	F	20.7	24.1
Greedy	T	F	23.6	49.7
Greedy	F	F	22.2	40.7

Refitting sense-embeedings allows us to know the sense

- New embeddings are defined as a as a weighted sum of unsupervised embeddings.
- The weights are determined using the langauge model, with a example sentence.
- This lets us find embedding for the sense of the word in that sentence.

$$\overbrace{\overbrace{\texttt{wow}}^{w_1}\ \overbrace{\texttt{the}}^{w_2}\ \overbrace{\texttt{wool}}^{w_3}\ \overbrace{\texttt{from}}^{w_4}\ \overbrace{\texttt{the}}^{w_5}\ \ \underbrace{\texttt{kid}}_{\textit{target word}}\ \ \overbrace{\texttt{is}}^{w_6}\ \overbrace{\texttt{so}}^{w_7}\ \overbrace{\texttt{soft}}^{w_8}\ \overbrace{\texttt{and}}^{w_9}\ \overbrace{\texttt{fluffy}}^{w_{10}}}^{\mathbf{c}}$$

Refitting sense-embeedings allows us to know the sense

- New embeddings are defined as a as a weighted sum of unsupervised embeddings.
- The weights are determined using the langauge model, with a example sentence.
- This lets us find embedding for the sense of the word in that sentence.
- Applications for similarity with context, and lexical tasks, such as Word Sense Disambiguation.

$$\overbrace{\overbrace{\texttt{wow}}^{w_1}\ \overbrace{\texttt{the}}^{w_2}\ \overbrace{\texttt{wool}}^{w_3}\ \overbrace{\texttt{from}}^{w_4}\ \overbrace{\texttt{the}}^{w_5}\quad \underbrace{\texttt{kid}}_{\textit{target word}}\quad \overbrace{\texttt{is}}^{w_6}\ \overbrace{\texttt{so}}^{w_7}\ \overbrace{\texttt{soft}}^{w_8}\ \overbrace{\texttt{and}}^{w_9}\ \overbrace{\texttt{fluffy}}^{w_{10}}}^{\mathbf{c}}$$

References

Eneko Agirre et al. “Evaluating and optimizing the parameters of an unsupervised graph-based WSD algorithm”. In: *Proceedings of the first workshop on graph based methods for natural language processing*. Association for Computational Linguistics. 2006, pp. 89–96.

Sergey Bartunov et al. “Breaking Sticks and Ambiguities with Adaptive Skip-gram”. In: *CoRR* abs/1502.07257 (2015).

Eric H Huang et al. “Improving word representations via global context and multiple word prototypes”. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics. 2012, pp. 873–882.

George A Miller. “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11 (1995), pp. 39–41.