Two/Too Simple Adaptations of `word2vec` for Syntax Problems

- *Wang Ling et al*.

Shubhangi Agarwal (14111268)

About word2vec

- ➔ proposed by Mikolov et al. 2013
- ➔ one of the most widely used tools for word vectors
- ➔ efficient implementation of **two** models
  - ◆ continuous skip-gram
  - ◆ continuous bag-of-words
- ➔ both models discard word order information
- ➔ embeddings built are suboptimal for syntax-based tasks
- ➔ e.g. pos tagging, dependancy parsing

In this work . . . {wang2vec}

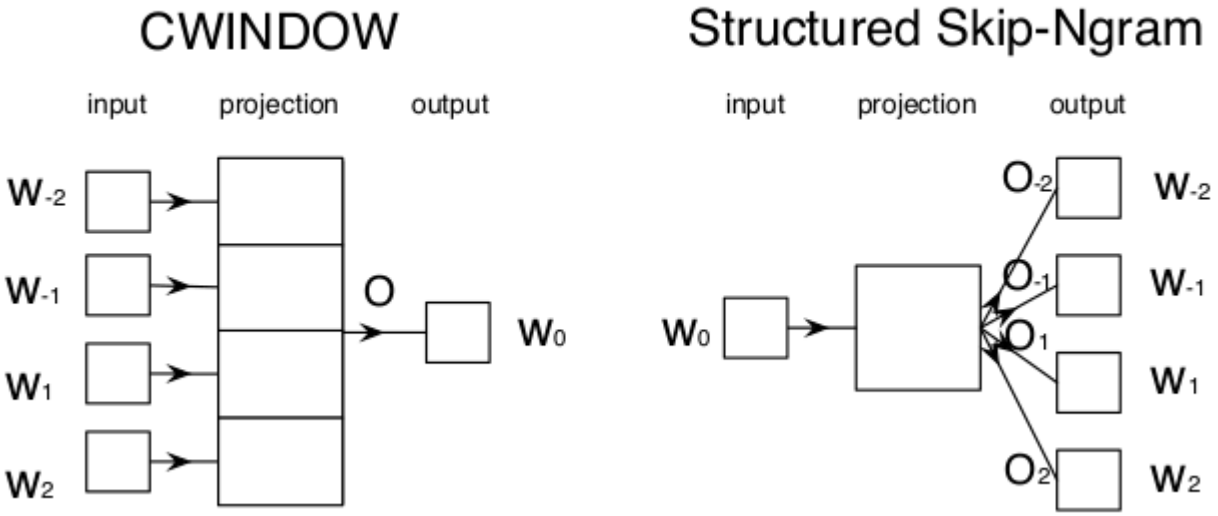
- two simple modifications to `word2vec`,
  - one for the skip-gram model
  - one for the CBOW model
- improve quality of embeddings for syntax-based tasks
- **code** : https://github.com/wlin12/wang2vec
- **goal** : to improve final embeddings while maintaining simplicity and efficiency of original models
- **proposed models:**
  structured skip-gram and *continuous window*

In this work . . .

- ➔ demonstration of the effectiveness of these approaches by training, on commodity hardware, on datasets containing more than 50 million sentences and over 1 billion words in less than a day,
- ➔ shown that these methods lead to improvements when used in state-of-the-art neural network systems for **part-of-speech tagging** and **dependency parsing**, relative to the original models
- ➔ both proposed models, increase the number of parameters of matrix O by a factor of $c \times 2$, which can lead to sparcity problems when training on small datasets. However, these models are generally trained on datasets in order of 100 millions of words, where these issues are not as severe

Structured word2vec

illustration of ***continuous window*** and ***structured skip-gram***



Structured Skip-gram

- **skip-gram model** uses a single output matrix $\mathcal{R}^{|V| \times d}$ to predict every contextual word $O \in w_{-c}, ..., w_{-1}, w_1, ..., w_c$, given the embeddings of the center word w_0
- proposed approach adapts the model for word-position sensitivity
- defines set of $c \times 2$ output predictors $O_{-c} .. O_{-1}, O_1 .. O_c$ with size $O \in \mathcal{R}^{|V| \times d}$
- each of the output matrices dedicated to predicting output for a specific relative position to the center word
- when making a prediction $p(w_o | w_i)$, select the appropriate output matrix O_{o-i} to project word embeddings to output vector
- number of operations that must be performed for forward and backward passes in the network remains the same as since simply switching the output layer O for each different word index

Continuous Window Model

- **CBOW** words model defines a window of words $w_{-c} \dots w_{c}$ with size c, where the prediction of the center word w_0 is conditioned on the remaining words $w_{-c} \dots w_{-1}, w_1 \dots w_c$
- prediction matrix $O \in \mathcal{R}^{|V| \times d}$ is fed with sum of embeddings of the context words. (order of the contextual words does not influence the prediction)
- **proposed approach** defines a different output predictor $O \in \mathcal{R}^{(|V| \times 2cd)}$ which receives as input a $(2c \times d)$-dimensional vector that is concatenation of embeddings of context words in the order they occur $[e(w_{-c}) \dots e(w_{-1}), e(w_1) \dots e(w_c)]$
- matrix O defines a parameter for the word embeddings for each relative position, this allows the words to be treated differently depending on where they occur

Experiments: Building Vectors

Most similar words using different word-embedding models

Embedding	**WIKI (S)**	**Twitter**	**WIKI (L)**
query	*breaking*	*amazing*	*person*
CBOW	breaks, turning, broke break, stumbled	incredible, awesome, fantastic, phenomenal, awsome	someone, anyone, oneself, woman, if
Skip-gram	break, breaks, broke, down, broken	incredible, awesome, fantastic, phenominal, phenomenal	harasser, themself, declarant, someone, right-thinking
CWindow	putting, turning, sticking, pulling, picking	incredible, amaaazing, awesome, amzing, a-mazing	woman, man, child, grandparent, servicemember
Structured Skip-gram	break, turning, putting, out, breaks	incredible, awesome, amaaazing, ah-mazing, amzing	declarant, circumstance, woman schoolchild, someone

Experiments: POS tagging

Results of POS tagging on PTB and Twitter datasets.

Cells indicate part-of-speech accuracy of each experiment

	PTB		**Twiter**	
	Dev	Test	Dev	Test
CBOW	95.89	96.13	87.85	87.54
Skip-gram	96.62	96.68	88.84	88.73
CWindow	**96.99**	97.01	**89.72**	89.70
Structured Skip-gram	96.62	**97.05**	89.69	**89.79**
SENNA	96.54	96.58	84.96	84.85

Experiments: Dependency Parsing

Results of dependency parsing on PTB using various models.

UAS: unlabelled parsing score, **LAS**: labelled attachment score

	Dev		**Test**	
	UAS	LAS	UAS	LAS
CBOW	91.74	88.74	91.52	88.93
Skip-gram	92.12	89.30	91.90	89.55
CWindow	92.38	89.62	92.00	89.70
Structured Skip-gram	**92.49**	**89.78**	**92.24**	**89.92**
SENNA	92.24	89.30	92.03	89.51

Conclusion

- ➔ two modifications to the original models in `word2vec` that improve the word embeddings obtained for syntactically motivated tasks
- ➔ by introducing changes that make the network aware of the relative positioning of context words.
- ➔ improvements in two mainstream NLP tasks, namely part-of-speech tagging and dependency parsing
- ➔ results generalize in both clean and noisy domains

References

- Wang Ling, et al. 2015. **Two/Too Simple Adaptations of word2vec fo Syntax Problems,** *NAACL 2015.*

- Tomas Mikolov, et al. 2013. **Distributed representations of words and phrases and their compositionality**. In Advances in *Neural Information Processing Systems*, pages 3111–3119.