Detecting Syntactic Change Using a Neural POS Tagger

by Merrill et al. [2019]

6 LP Seminar "Diachronic Language Models"

Maya Arseven

Heidelberg University
Institute of Computational Linguistics
arseven@cl.uni-heidelberg.de

January 9, 2024

Outline

Outline

Definition

Syntactic Change

Changes in a language that occur on syntactical level

Ex.: Change from OV word order in Old English to VO order in Middle and Modern English

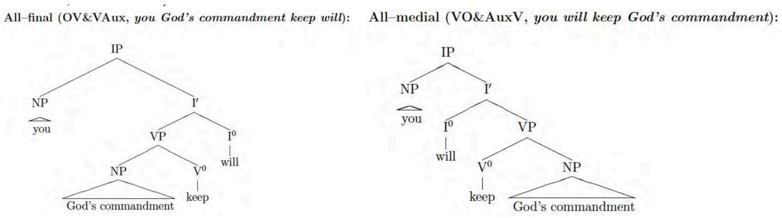


Figure 1: Word order change from Old in Modern English, (Flemming [2019])

Motivation

- Prior work had more focus on semantic change (Dubossarsky et al. [2017], Hamilton et al. [2016], Jo et al. [2017])

$\rightarrow$ The authors wanted to analyze the **syntactical** change in modern American English

Motivation

- Prior work had more focus on semantic change (Dubossarsky et al. [2017], Hamilton et al. [2016], Jo et al. [2017])

→ The authors wanted to analyze the **syntactical** change in modern American English

- The ones focusing on syntax used other techniques: Niyogi and Berwick [1995] developed a mathematical simulation based on language contact and acquisition

→ The authors wanted to learn about syntactical change with the help of **neural networks**

The Idea

1. Build an **LSTM POS tagger**
   - Assign POS tags to sentences dating from a specific year
   - Learn about temporal progression in the process
2. Extract and analyze the learned **year embeddings**
   - Reduce dimensionality with PCA
   - Calculate the correlation to time and determine the cause
3. Perform **temporal prediction** with the learned year embeddings
   - Bucket the years into decades
   - Predict the creation date of given sentences

Outline

COHA

Davies [2010]

- Corpus of Historical American English (COHA)
- Documents dating from 1810 to 2009
- Balanced by genre: Fiction, academic, newspapers, magazines
- Balanced by decade: Randomly selecting 50k sentences from each decade
- Total of 1M sentences after pre-processing
- 90/10 train-test split
- Reduced to max length of 50 words
  - Loss of 7% of the data $\rightarrow$ 70k sentences
  - The analysis on long dependencies may be lost as well

COHA

- Annotated with word, lemma and POS information
- Least-specific tag model is selected with 423 unique POS tags

CC	Coordinating conj.	TO	infinitival *to*
CD	Cardinal number	UH	Interjection
DT	Determiner	VB	Verb, base form
EX	Existential there	VBD	Verb, past tense
FW	Foreign word	VBG	Verb, gerund/present pple
IN	Preposition	VBN	Verb, past participle
JJ	Adjective	VBP	Verb, non-3rd ps. sg. present
JJR	Adjective, comparative	VBZ	Verb, 3rd ps. sg. present
JJS	Adjective, superlative	WDT	Wh-determiner
LS	List item marker	WP	Wh-pronoun
MD	Modal	WP$	Possessive *wh*-pronoun
NN	Noun, singular or mass	WRB	Wh-adverb
NNS	Noun, plural	#	Pound sign
NNP	Proper noun, singular	$	Dollar sign
NNPS	Proper noun, plural	.	Sentence-final punctuation
PDT	Predeterminer	,	Comma
POS	Possessive ending	:	Colon, semi-colon
PRP	Personal pronoun	(	Left bracket character
PP$	Possessive pronoun	)	Right bracket character
RB	Adverb	"	Straight double quote
RBR	Adverb, comparative	‘	Left open single quote
RBS	Adverb, superlative	“	Left open double quote
RP	Particle	’	Right close single quote
SYM	Symbol	”	Right close double quote

Figure 2: As comparison, The Penn Treebank has 36 unique POS tags, (Taylor et al. [2003])

Word Embeddings

Mikolov et al. [2013]

- Pre-trained 300-dim Google News word embeddings, also used in `word2vec`
- In the case of OOV $\rightarrow$ Assign a vector from the normal distribution of the vocab so that every OOV word gets different embeddings
- Filtering to the top 600k words, others are marked as `UNK`

Q: Is this a problem considering a lot of rare words (which a lot of Old English words are) would be marked as `UNK`?

Word Embeddings

Mikolov et al. [2013]

- Pre-trained 300-dim Google News word embeddings, also used in `word2vec`
- In the case of OOV $\rightarrow$ Assign a vector from the normal distribution of the vocab so that every OOV word gets different embeddings
- Filtering to the top 600k words, others are marked as `UNK`

Q: Is this a problem considering a lot of rare words (which a lot of Old English words are) would be marked as `UNK`?

- Actually no, because the focus of the research is the **syntactic structure** of sentences and not lexical change.

Outline

1. Network Architecture

1. Build an **LSTM POS tagger**
   - Assign POS tags to sentences dating from a specific year
   - Learn about temporal progression in the process

1. Network Architecture

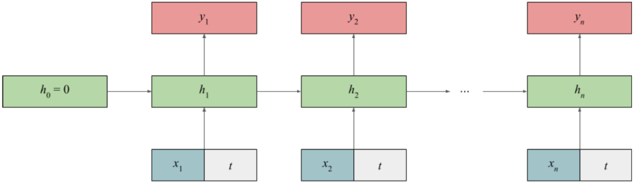


Figure 3: The single layer LSTM architecture with a size of 512, (Merrill et al. [2019])

- Input: x_i, t = [current word embedding, doc's year embedding]
- Output: y = predicted **POS tag** for the current word

1. Network Architecture

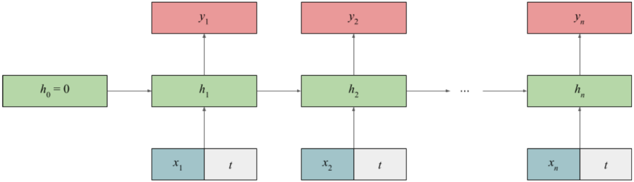


Figure 3: The single layer LSTM architecture with a size of 512, (Merrill et al. [2019])

Ex.: $s =$ `I have a cute cat`, $x_i =$ `cute`, $t = 2002$

- Input: x_i, $t =$ `[current word embedding, doc's year embedding]` `= [embedding for cute, embedding for 2002]`
- Output: $y_i =$ predicted **POS tag** for the current word $=$ **ADJ**

1. Network Architecture

- Representations for multiple decades are learned by a **single** model **dynamically** $\rightarrow$ not separate models for each decade
- So in *temporal prediction*, information on all years is used in the decision

- Additional to the main model, several ablation models:
  - LSTM without a year input
  - FF tagger with year input
  - FF tagger without year input

2. Analyzing Year Embeddings

2. Extract and analyze the learned **year embeddings**
   - Reduce dimensionality with PCA
   - Calculate the correlation to time and determine the cause

2. Analyzing Year Embeddings

- Reduce the 300 dim year embeddings into 1D using PCA
- Calculate the correlation between the first PC and time both for FF and LSTM models

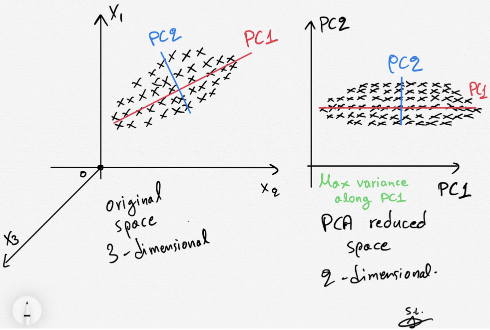


Figure 4: How PCA works, (Serafeim [2020])

2. Analyzing Year Embeddings

- To understand the cause of this correlation compare LSTM and FF models
- Both models capture lexical info but due to its **recurrent connections** LSTM also encodes syntactical info
  - LSTM is a type of RNN with memory advancements

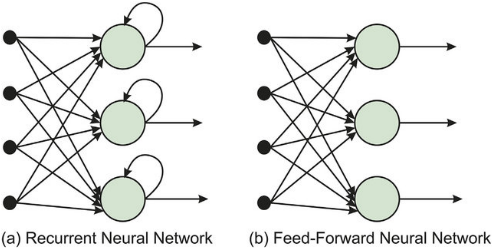


Figure 5: The comparison between RNN and FFNN, (Eliasy and Przychodzen [2020])

3. Temporal Prediction

3. Perform **temporal prediction** with the learned year embeddings
   - Bucket the years into decades
   - Predict the creation date of given sentences

3. Temporal Prediction

- Bucket the years into decades, predicting the exact year could be too hard
  - Later confirmed in the results
- Evaluate the models ability to predict the composition year of given sentences
- Compare the predictive year to the gold year

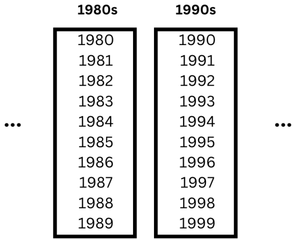


Figure 6: Decade bucketing illustration, (created by me)

Outline

Research Questions

- `RQ1:` Is temporal progression encoded in the networks learned year embeddings?
- `RQ2:` Does the represented temporal change reflects syntax rather than simply word frequency?
- `RQ3:` Can the model be used to date novel sentences?

Tagger Performance

	Feedforward	LSTM
Year	82.6	**95.5**
No Year	77.8	95.3

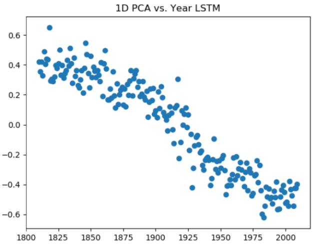
Table 1: Accuracies on test sets for each model

	Feedforward	LSTM
Year	82.6	**95.6**
No Year	77.7	95.4

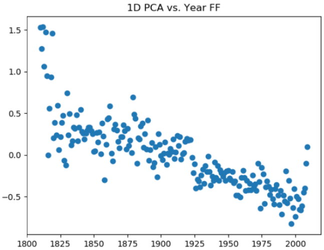
Table 2: Accuracies on train sets for each model

- A verification of a functioning tagger is sufficient
  - Main goal is the *temporal prediction*
- No sign of overfitting
  - Accuracies are comparable on train and test sets
- Overall, the performance is worst in FFs because they don't take the relations between words into consideration

Analyzing Year Embeddings



Graph 1: The correlation of the 1. PC with time ($R^2 = 0.89$)

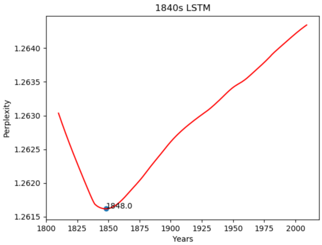


Graph 2: The correlation of the 1. PC with time ($R^2 = 0.68$)

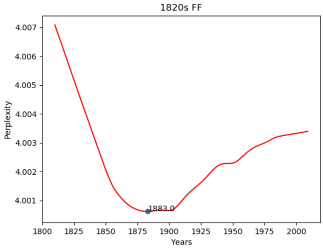
- LSTM: Clear linear relationship between first PC and time
  - 1. PC corresponds to the order of years $\rightarrow$ Learning successful
- FF: A weaker correlation

Temporal Prediction

- `LOWESS` curve: The model's certainty for each prediction, whereas the minima corresponds to the predicted label
  - Measured in perplexity (measure of uncertainty), so the lower the better



Graph 3: The 1840s `LOWESS` curve for the LSTM. The predicted year is 1848.



Graph 4: The 1820s `LOWESS` curve for the FF. The predicted year is 1883.

Temporal Prediction

	Baseline	Feedforward	LSTM
Decade	50.0	26.6	**12.5**
Year	50.0	37.5	**21.9**

Table 3: Average distance between the predicted and actual years of composition

- **Baseline**: Predicting the middle year in the dataset (1910) for every sentence
- Decade bucketing with LSTM achieved the best performance
  - The model can be used to date novel sentences

Case Analysis

Sentence	Year Pred	Year True	Error FF	Error LSTM
1. it is of great consequence, that we adorn the religion we profess, and that our light shine more and more that we grow in grace as we advance in years, and that we do not resemble the changing wind or the inconstant wave.	1817	1817	86	0

Table 4: Case analysis with the 10 best predicted sentences by the LSTM

- S1 was predicted perfect, 0 error in LSTM but 86 error in FF
    - Syntax must have helped in the LSTM prediction

Outline

Conclusion

- **The LSTM model** was successful to date new sentences with correctly learned **year embeddings** in the **POS tagging** task
- The performance of the LSTM was stronger in comparison to the FF or baseline models
  - Because it could also capture **syntactical change**

Conclusion

- **The LSTM model** was successful to date new sentences with correctly learned **year embeddings** in the **POS tagging** task
- The performance of the LSTM was stronger in comparison to the FF or baseline models
  - Because it could also capture **syntactical change**

Future work:

- How to capture *continuous grammatical change*?
  - Aboh [2015]: Speakers always have multiple grammars available to them and they choose between them
  - By time one grammar becomes more prevalent

Final Words

What I liked about the work:

- An unique way to approach temporal prediction
- Not a complicated architecture $\rightarrow$ easy to understand

Final Words

What I liked about the work:

- An unique way to approach temporal prediction
- Not a complicated architecture $\rightarrow$ easy to understand

What I thought could be better:

- Limiting to max 50 words could lead to information loss
- Only one training epoch
- No comparison to results from other papers

Questions

Questions

Sources I

[1] E. O. Aboh. *The Emergence of Hybrid Grammars: Language Contact and Change*. Cambridge Approaches to Language Contact. Cambridge University Press, 2015. URL https://www.cambridge.org/core/books/emergence-of-hybrid-grammars/F61C39AC98FCE098A9140DB46E843DDA.

[2] M. Davies. The corpus of historical american english: 400 million words, 1810-2009. 2010. URL https://www.english-corpora.org/coha/.

[3] H. Dubossarsky, D. Weinshall, and E. Grossman. Outta control: Laws of semantic change and inherent biases in word representation models. In M. Palmer, R. Hwa, and S. Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1136–1145, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1118. URL https://aclanthology.org/D17-1118.

Sources II

[4] P. E. Flemming. Language variation and change - section 14: Syntactic change 2, 2019. URL https://ocw.mit.edu/courses/24-914-language-variation-and-change-spring-2019/resources/mit24_914s19_lec14/. As taught in Spring 2019. Accessed on Date.

[5] W. L. Hamilton, J. Leskovec, and D. Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. In K. Erk and N. A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1141. URL https://aclanthology.org/P16-1141.

[6] E. S. Jo, D. Shen, and M. Xing. Backprop to the future: A neural network approach to linguistic change over time. Manuscript, Stanford University, 2017. URL https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2761010.pdf.

Sources III

[7] W. Merrill, G. Stark, and R. Frank. Detecting syntactic change using a neural part-of-speech tagger. pages 167–174, Aug. 2019. doi: 10.18653/v1/W19-4721. URL https://aclanthology.org/W19-4721.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013. URL https://arxiv.org/abs/1301.3781.

[9] P. Niyogi and R. C. Berwick. The logical problem of language change. AI Memo 1516, Artificial Intelligence Laboratory, MIT, 1995. URL https://dspace.mit.edu/bitstream/handle/1721.1/7196/AIM-1516.pdf?sequence=2&isAllowed=y.

[10] A. Taylor, M. Marcus, and B. Santorini. The penn treebank: An overview. 01 2003. doi: 10.1007/978-94-010-0201-1_1. URL https://www.researchgate.net/publication/2873803_The_Penn_Treebank_An_overview.