Exploring Future Work: Using Language Models to Detect Syntactic Change Over Time

Maya Arseven & Hiu Lam Choy
{arseven|choy}@cl.uni-heidelberg.de
Heidelberg University, Institute of Computational Linguistics
Seminar: Diachronic Language Models, WS 2024
Lecturer: Wei Zhao

Abstract

This paper provides first a comprehensive survey of different approaches in previous studies for detecting syntactic change over time and second a presentation of our vision for future work for deploying language models. Despite the field's spotlight on semantic change detection, we recognize the latent interconnection between syntactic and semantic changes. Therefore, we include a mix of the two aspects in our review. Our aim is to give an extensive overview of the current advancements in the field of syntactic change detection. Additionally, we present our own insights on the topic and propose several innovative ideas that could be valuable for future research.

1 Introduction

As the words of the Greek philosopher Heraclitus suggest, the only constant in life is change, a principle that applies even to language. Over time, language undergoes transformations at every structural level, phonology, morphology, syntax and semantics. These changes are not uniform across languages, dialects, or even within the different dimensions of a single language, leading to significant variation. For instance, a comparison of the syntax of English and Japanese from the medieval period to the present day reveals that English has undergone substantial changes, whereas Japanese syntax has remained relatively stable (Kroch, 2008).

The task of detecting syntactic change over time is, in simple words, capturing any changes in a language that occur on a syntactical level. The scope of syntactic change can range from shifts in word order to the process of grammaticalization. A typical example (Figure 1) in this field is the change from OV word order in Old English to VO order in Middle and Modern English.

Capturing such changes can be particularly challenging because of the complex nature of the task. It requires expert knowledge to understand the syntactical structure in multiple time frames as well as to separate syntactical change to focus on that information only. Although there were many attempts to investigate into semantic or lexical change (Schlechtweg et al., 2019; Hu et al., 2019; Giulianelli et al., 2020; Ma et al., 2024), the research on detecting syntactic change over time with the help of language models (LMs) seems to be still lacking on many aspects. The aim of this paper is to take a critical and creative approach to explore future directions on using LMs to detect syntactic change over time with the foundation from previous work on detecting semantic change.

In this paper we propose several novel ideas hoping to inspire future researchers in their work on detecting syntactic change over time with the help of LMs. We first take a deep dive in previous work on detecting semantic and syntactic change in section 2. Utilizing what we have learned from the previous work, we then address open issues on the field in section 3. Then comes the main body of our paper, where we propose our vision for future work in section 4. Although we try to cover everything, we discuss our shortcomings in section 5 and conclude our work in section 6.

2 Background

In the field of diachronic language models, detecting semantic change has certainly gained more attention, whereas syntactic change has been rather understudied. This may be attributed to the greater impact of investigating the meaning shifts revealing more about the evolution of a language then structural change. Syntactic changes should be more straight-forward to detect as they are more low-level features than the hidden meaning behind a word. For that reason, most researches put their focus and effort on detecting the semantics, hoping their research to bring a bigger impact to the field. We believe that there are no strict boundary between the both aspects, but rather a continuum.

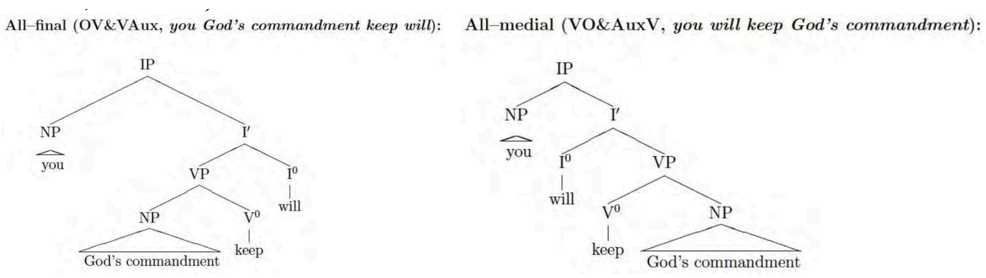

Figure 1: Word order change from Old in Modern English (Flemming, 2019)

That is why in this section, we will be first exploring prior work in the neighboring topic of semantic change in the hopes for getting inspiration for future work on detecting syntactic change with the help of LMs. Then, we will look into some previous works on detecting syntactic change, which have brought noteworthy contributions to the field. And at last, we mention the first works that utilized LLMs for detecting language change.

2.1 Detecting Semantic Change

Distributional Semantic Models As the British linguist J.R. Firth once said this well-known quote: "You shall know a word by the company it keeps", the theory of distributional semantics proposes that words occurring similar context would also carry similar meaning. A distributional semantic model (DSM) has the objective to capture the semantics of a word in a given corpus, while at the same time indirectly captures the syntactic structure of the text. There are two types of DSMs, to be specific, the count and predictive model. (Baroni et al., 2014) Count models use co-occurrence count matrices to represent the relationship between the word and the corpus. In contrast, predictive models use word embeddings generated by techniques like Word2Vec to capture the semantics of a word in the corpus. The study conducted by Fonteyn et al. once again fuses the detection of semantic and syntactic changes through the use of distributional semantic models. Their models have the ability to detect morphosyntactic constructions that have undergone functional-semantic change. For instance, the change of the word *must* from just having a deontic sense *"you must listen!"* to additionally having an epistemic sense *"the bread must have been stale"*. The fundamental syntactic parses the model made can be useful to predict abstract syntactic structures that are harder to be detected, which could serve as extra information for the task of syntactic change detection.

Contextualized Methods Kutuzov et al. evaluated various contextualized methods for detecting semantic changes. These contextualized methods can be integrated into different architectures and create contextualized tokens embeddings for the task. Specifically, they used the inverted cosine similarity over word prototypes (PRT) and the average pairwise cosine distance (APD). On the one hand, PRT computes the change score by calculating the inverted cosine similarity between the average of all embeddings of a token in two different time periods. On the other hand, APD measures the average distance between all token embeddings in two different time periods. Both methods they used compute a change score from the embeddings, which correspond to the degree of semantic change the words have undergone. Kutuzov et al. discover that averaging both the outcomes from PRT and APD always produces a better performance than just using either one. This is especially advantageous when the gold score distribution is not known. They subsequently incorporated these methods to BERT and ELMo models and did a comparison between them, and their discovery shows that integrating the PRT and APD methods into ELMo outperforms BERT slightly.

However, the results are still not totally convincing. Other than detecting some unwanted changes which are not necessarily of the semantic changes expected by a human, one of the three main problems that they encountered is the misclassification of the model's prediction of the presence of semantic change when the word is actually undergoing syntactic change. Kutuzov et al. propose an idea of using a multilayer LM instead, as the lower layer carries more syntactic information and the upper layer more semantic information, as shown in the work of Peters et al., to differentiate the various kinds of language change.

Temporal Generalization of LMs Since pretrained LMs (PLMs) are trained on static data from the past, they often perform poorly on temporal generalization tasks. They are trapped in the time-agnostic setting in which they were trained. Since one of the main applications of PLMs is fine-tuning with new and recent data to perform downstream tasks, the models often encounter temporal misalignment. PLMs, such as BERT and RoBERTa, can be further trained with recent data. Nonetheless, this is firstly much costly, and secondly results in the model forgetting its initial data. In an attempt to temporally adapt PLMs, Su et al. research on the correlation between lexical semantic shifts in the PLM training data and its temporal generalization capabilities. Their results suggest that focusing on the words that have undergone semantic changes in a masked language model (MLM) setting contributes much more to mitigating the temporal alignment problem than relying on typical random maskings of an MLM. That is why they propose a lexical-based MLM (LMLM) to capture lexical semantic change between different temporal splits.

2.2 Detecting Syntactic Change

Probabilistic Approaches To delve deep and get the latent structural information of the data, an approach of using an infinite relational model (IRM) is investigated by Hellwig and Sellmer, where the model is able to find similar constituents and bias terms that are most likely to be the critical factors of the model's decision. The chronological information is integrated in the nodes of a similarity graph representing the constituents, where the distribution of the constituent over time can be evaluated. If both constituent forms appear at least two times in the dataset, a binary relationship between the two constituents will be established, helping us to capture their syntactic similarities. The IRM model can group partitions with constituents that contain similar syntactic information together, which helps to prioritize the most relevant and influential constituents that possibly drive syntactic changes, thereby simplifying the processing of data.

In another approach to investigate into syntactic change, Zampieri et al. use Multinomial Naive Bayes models (MNBs) to classify texts from different time periods, and then evaluate the features that contribute to the model's decision. They discover the potentials of integrating N-grams into the model, where using a trigram model captures the structural differences between sequences after converting the tokens into part-of-speech (POS) tags. It works successfully as sequences should be considered when investigating into syntax, while a unigram might be more suitable for looking into the semantic and word-based information due to the consideration of the word itself. This is a simple approach yet works surprisingly well on the task of detecting syntactic change.

Support Vector Machines Using the same principle of finding the boundary that set different periods apart, Zampieri et al. also use a linear support vector machine (SVM) classifier for the same task as the previous subsection as a comparison, and this approach is shown to be more accurate than using MNBs. Features that contribute to the models' decision can be easily detected by comparing the lexical and morpho-syntactic information of the texts. The weights assigned by the SVMs show the importance of the feature in contributing into the decision. By ranking the weights, the most informative feature can be detected. Zampieri et al. also experimented on using texts with different features, namely the words, POS tags and a mixture of both in a bag-of-words representation. Their results show the presence of both words and POS tags achieve a higher accuracy than only using either one of the features, as the varied linguistic features might have provided more information for the model to understand the relationship between lexicons and grammar better, meaning a combined investigation of semantics and syntax enhances the detection of period-determining syntactic features.

In another study, Dunn and Wong also implemented syntactic models as SVMs in a different task on dialect classification models, where they have the ability to analyse the syntactic variation across space and time. They started by constructed a set of grammar using the construction grammar (CxG), which is discussed in section 4.4, and use that to formulate syntactic representations of each of the investigated dialects of English. With the bag-of-constructions approach, the occurrence of each constructions are counted and used as a feature space to carry out the task of dialect classification. By testing their models with datasets from different periods, an indication of syntactic change can be detected if the model's performance decreases.

Additionally, Dunn and Wong used a regression analysis to model the decay rate of their models. Their theory suggests that a consistent decay rate indicates that the model itself has made the error, while a faster decay rate indicates the inability of the model to catch up on the changes occurring in the language. By examining the change in F-score over time, the robustness of the models across different time periods can be evaluated, allowing us to approximate the extent of changes the language has undergone. While their method detects the presence of syntactic changes, the location of the change itself remains elusive.

Neural Language Models Merrill et al. present one of the first works that uses a LM to detect syntactical change. They make use of a single-layer LSTM to detect syntactic change. In comparison to feed-forward (FF) networks, the LSTM architecture allows the encoding of syntactical information due to its recurrent connections, whereas FFs fall short by only capturing lexical information. The main goal of their work is to predict the origin year of novel sentences, but to do so, vector representations for each year or decade are needed. For that reason, the authors first start by training a neural POS tagger. And by training this LSTM POS tagger, yearly embeddings are gained dynamically in the process. The embeddings represent the synactical structure for each year. Then, to deploy these embeddings and make sure that what they have learned is correct, they analyze the embeddings by performing dimensionality reduction techniques and investigating their correlation with time. When they finally perform temporal prediction with the LSTM model and the year embeddings, their results show that they surpass their baselines. This indicates that they can successfully predict the composition year of novel sentences with their model.

BERT-based Models Hou and Smith uses a fine-tuned BERT model to distinguish between texts from 1800s and 1900s. The model can achieve this by using syntactical information only, meaning that the semantic information have not been made available to the model. Later, the authors uses the same fine-tuned BERT model to identify syntactical changes in corpora, such as the introduction of a new POS tag. This requires no manual annotations and runs completely in an unsupervised fashion. The old text is not being passed through an automatic POS-tagger, as the result could be unreliable. Instead, they make use of the plain old text with the tagged modern text.

The fine-tuned BERT model gets an additional linear layer to predict the origin year of a sentence. In order to make sure that the model utilizes syntactical information only, they follow a similar approach to Gulordava et al.. They replace the actual words in the sentence with their POS tags and then replace these again with common words in the respective POS tag categories. The transformation of a sentence can look like this: *"ChatGPT is a popular tool amongst students"* $\rightarrow$ *"NNP VBZ DT JJ NN IN NNS"* $\rightarrow$ *"Ashley walks the confused apple with ideas"*. Another feature that can influence the prediction is the sentence length. The authors also carried out experiments with fixed-size sentence lengths and confirmed that the length doesn't play a significant role in the prediction. So the model learns the syntactical differences between 1800s and 1900s. The question now is to how to detect these changes.

The authors train first two separate POS taggers with BERT, one with 1800 and the other with 1900 data. Then they save both embeddings for each word that appears in the 1900s data, as they say this would be more reliable. And they train a perceptron to assign POS-tags to each embeddings. They experiment with two approaches to detect where the syntactic change occurs. First they find the words that the two models disagree while assigning a POS-Tag and second masking out the target word to see if the 1800s model predicts the tag correctly. The results show that the masking approach performs better.

2.3 Large Language Models

Since October 2023, language models are not just a research field of Computational Linguistics but also part of the common crowd's daily life. It has shown to be successful in many tasks such as summarization, question-answering and many more. But one of the main disadvantages of a large language model (LLM) is that it is static or have been frozen after it's training has been completed, which is the reason behind the famous sentence *"My responses are based on the information available to me up until September 2021."*. This quality makes it particularly challenging for deploying LLMs into diachronic tasks, since they require a dynamic information flow from the nature of the task.

Wang and Choi's 2023 paper was the only work that we found so far that dealt with the intersection of LLMs and detecting language change. There is a lot of room for exploration, since it is proven that LLMs can bring unexpected success in zero-shot settings or with few-shot prompting techniques.

3 Open Issues

After having a thorough look at the domain of detecting syntactic change, several challenges seems to remain unaddressed or unresolved.

3.1 Types of Syntactic Change

Almost all the prior work deals with one type of syntactic change, which is connected to POS tags and their order. It is either the change of a POS tag or an introduction of a new tag. But there are also many instances in syntactic change where the POS tag stays the same with only its morphology changed.

For example, in cases of grammaticalization, where a word changes from being less grammatical to more grammatical, the POS tag of a word may remain unchanged, although there is a syntactical shift happening (Roberts and Roussou, 2007). The phrase "going to" had originally the meaning of the act of moving towards a place. Over time it has evolved into a future tense marker, also expressed informally as "gonna." This transformation narrowed the syntactic flexibility and placement. Initially, "going to" could be used in multiple syntactic structures, such as describing physical movement *"I am going to the store"* or indicating future plans *"We are going to the beach tomorrow"*. However, as "going to" became a future marker, its usage narrowed down to denote futurity *"I am going to watch a movie tonight"*. This restriction led to positioning directly before the base form of the verb it modifies, with its syntactic role tightly bound to expressing future actions or plans.

Also formation of new compound words, such as makeup and make up, can alter the POS sequence in a sentence. With all these various kinds of syntactic change, it is challenging to have one robust model able to detect every single change.

3.2 Data Availability

One of the main limitations that most studies encounter is the availability of data sets. A large amount of training data from different periods is required to make the model familiarized with the writing styles of each period, and this is especially more challenging for historical periods. Many ancient documents are not digitalized for processing yet, thus some extra processing might be required before the data become readily usable.

3.3 Deciding on the Intervals

Choosing the intervals between training and testing periods is a crucial factor in determining the performance of the models. The interval size essentially depends on the characteristics of the dataset we are using; If the dataset spans across a long period of time, it would be sensible to employ larger intervals as we can have a better insight of an overall trend of the changes occurred. Whereas using smaller intervals can precisely locate the occurrence of the changes, helping us to understand syntactic change on a granular level in spite of the higher computational costs and time. Ultimately, experimenting the intervals and adjusting them to our fit will be the best way to find out the optimal boundary,

3.4 Evaluation of the models

The task of syntactic change detection is usually done in an unsupervised approach; the goal is to generalise patterns from the data itself rather than comparing it to a gold standard. Typically, there are only limited labelled datasets available openly for such tasks. Should we desire to use annotated data, expert annotations are required, which in turn demand additional time and human effort to obtain. As observed, most models for syntactic change detection often identify some linguistic phenomenon as a syntactic change which does align with our human judgement. For that reason, it is still expensive to evaluate the results of the models without human assessment.

4 Our Vision for Future Work

In the following section, we will combine our learnings from previous work and open issues in the domain with the hopes of proposing novel ideas for future work on using LMs to detect syntactic change over time.

4.1 Syntactic-based MLMs

One possible approach could be to build a **syntactic-based masked language model** (SMLM) to detect syntactic change over time, like how Su et al. did with a lexical-based one to detect semantic shifts on a lexical-level.

SMLMs, such as Bai et al.'s SyntaxBERT, are nothing new. But to the best of our knowledge, they have never been tailored for diachronic settings before. SyntaxBERT is built on the widely used Devlin et al. BERT model by adjusting the conventional self-attention layer and attention masks to leverage syntactical information. The authors add a parent and child mask as well as a sibling mask, denoting the relationship of tokens in a dependency tree structure in these different ways. The additional topical attention layer built on top of the standard self-attention is supposed to learn which of the subnetworks, hence, which of the maskings are most important for the task.

SyntaxBERT is a good starting point for building a SMLM explicitly for detecting syntactic change, similar to Su et al.'s work. Contrary to the traditional random or SyntaxBERT's masking strategy, our SMLM would preferentially masks the words that are signaling that they have gone through syntactic change over time. But it is important to keep in mind that the SMLM comes not into play until the last step of the pipeline, meaning there s additional components to adjust when defining the objective.

This would mean transforming the data into parse trees with POS-tag information and masking them according to different metrics. But since detecting syntactic change can be more complex, the first step of the pipeline, candidate word (in our case, sentence) selection would need adjusting.

4.2 Deploying Dependency Trees

A metric that measures syntactic complexity is the dependency distance between the constituents in a sentence. Lei and Wen argue that dependency distance faces minimization over time due to the limitations of human working memory. In a newly published paper, Chen et al. analyze minimum and normalized mean dependency distances as well as question the reliability of the commonly used Stanford CoreNLP parser (Manning et al., 2014a). Even though diachronic dependency trees have been used in the mentioned works, feeding them to a LM to detect syntactical change has not yet been explored to our knowledge. Considering the POS-tags in a sentence as a dependency-rooted tree, shown on the right in Figure 2, these sentences could be fed to a neural LM or a PLM, such as BERT, to detect syntactic change.

4.3 Zero-shot Prompting LLMs

Wang and Choi explored the use of LLMs to detect lexical change with a zero-shot setting, but to our knowledge, there is no work on detecting syntactic change while deploying LLMs.

Transforming their approach to detect syntactic change is relatively straight-forward. Their context prompt can be used without making alterations, and only the actual prompt would need a few modifications. Since we couldn't change the context prompt without an OpenAI API, we have decided to just do a small test with a simple prompt on ChatGPT4. With a little prompt engineering, we found out that the prompt shown in Figure 3 seemed to deliver desirable results in our naive experiment.

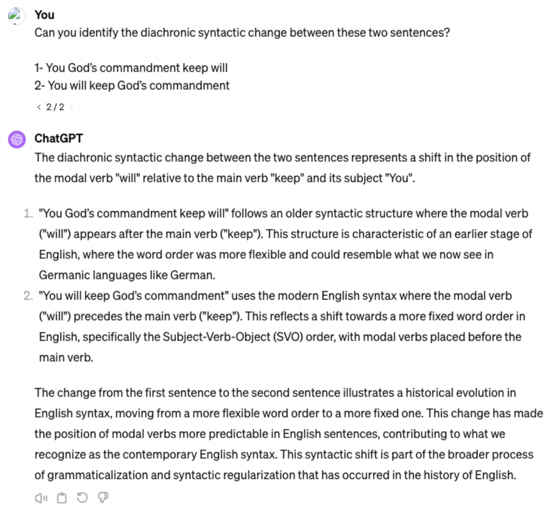

Figure 3: Experiment with ChatGPT4, see the chat here

4.4 Using Construction Grammar

Evaluating syntactic change should not be a standalone task but should also be complemented with additional linguistic analysis. Semantics and pragmatics often play a huge role in language change as well, which in turn influences the syntax of the sentence. Construction grammar (CxG) can possibly be applied to the syntactic detection model like in the study done by Dunn and Wong to create a grammar for the models. To create a set of CxG, we need to gather information for the lexical, syntactic and semantic constructions. As the lexical constructions, we can extract the most frequent N words across the corpus; For the syntactic constructions, we need to translate the tokens into POS tags; Lastly for the semantic constructions, we can, for instance, use fastText embeddings, just like how Dunn and Wong did, to capture the seman-

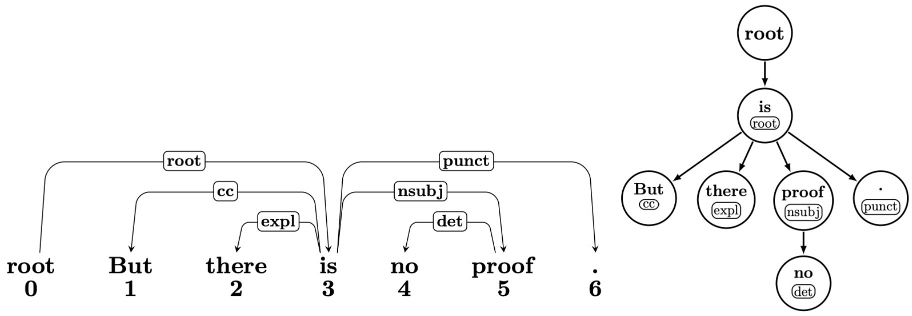

Figure 2: Dependency relations of sentence “But there is no proof.” in linear order (left) and tree graph (right). (Chen et al., 2024)

tic information of the words. Using the skip-gram method can filter out polysemies from the lexical constructions when defining semantic constraints, which could be practical. By integrating all these constructions together, we produce a set of CxG.

CxG believes that there is no strict separations between lexicon and syntax (Fried, 2015), but rather a continuum. It extends to the domain of semantics and integrates both syntactic and semantic features together, where it suggests the valency of a construction, meaning the number and type of arguments controlled by the verb, is influenced by the productivity of a verb. This productivity is essentially determined by the level of abstraction of a construction. We can additionally study the syntactic productivity of a construction using distributional semantics (Perek, 2016). They employed a vector-spaced model, generating vectors based on how often the words co-occur in the text, ultimately capturing similar vectors. As there will be many pairs of words with no relationship at all, the data would be sparsely distributed. Therefore, dimensionality reduction can be used to filter out the most outstanding and informative features. By identifying the similar constructions through multidimensional scaling and cluster analysis, we can predict possible emergence of a new construction when the similar constructions are seen to be undergoing changes in their syntactic structures.

CxG also tackles the problem of dealing with idioms and unusual expressions, which usually have uncommon syntactic structures. These less productive constructions could provide valuable insights to syntactic changes due to their relatively more stable nature, thereby enriching our understanding of the evolution of syntactic changes in a language.

4.5 Synthesizing the Datasets

To tackle the problem of limited datasets, we can generate artificial data by synthesizing sentences from different texts which originate from the same period. This approach has been done and proven successful by Zampieri et al., with the advantage of having a mixed style of texts as well as the ability to adjust the length of the data, which is especially helpful for the pre-processing steps to make all data comparable. Longer texts are also more beneficial to such tasks of syntactic analysis due to its richer contextual information, enabling an easier classification for the model.

4.6 Trankit

Due to a common usage of POS in the task, implementing them with the new state-of-the-art transformer-based toolkit Trankit[1] (Nguyen et al., 2021) could serve a beneficial purpose for processing languages. Trankit can be seamlessly integrated into different NLP pipelines for tasks such as POS tagging, dependency parsing, tokenization, and more. Furthermore, its support for a wide range of languages enables multilingual studies beyond English. Hou and Smith explored the utilization of Trankit, where they obtained POS and dependency tags from the library to train their models for detecting syntactic change. In addition, the authors performed a comparative analysis of its performance with the POS tags acquired from another widely-used NLP model, namely the Stanford CoreNLP library (Manning et al., 2014b). Their results revealed that the application of Trankit yields better results by 0.8% than the CoreNLP library, indicating its potential as an alternative NLP toolkit for future researches.

[1] `https://github.com/nlp-uoregon/trankit`

5 Discussion

Although we tried to cover the open challenges on the field mentioned in the section 3, there are still topics we weren't able to cover due to time limitations. Firstly being the boom of interest and research going into generative AI and LLMs in the last 3 years. Due to this current state of our field, it is challenging to keep up with what is being released every day; Some papers get outdated even before they see the light of publication. Although the interaction between LLMs and humanities, where the topic of this paper falls, seems to be unexplored so far, the future still remains unpredictable.

As far as what we have presented so far it is hard to offer one singular perfect approach to tackle the task. There are multiple components that play hand-in-hand while setting up a research and our aim is mainly to inspire future work tailored to their needs as they might tackle similar problems. For instance if one is focusing on deploying LLMs on their research they might utilize our zero shot prompting idea (section 4.3) or if they want to explore dependency distance, they might find ideas in our dependency trees idea (section 4.2).

6 Conclusion

This paper explores a diverse range of methods in the domain of syntactic change detection, along with some techniques and tools that could be beneficial to the task. We also take note on the approaches that used semantic change detection, which can possibly have valuable contributions to syntactic change detection due to the interconnections between them. Each approach we have discussed in this paper addresses a different aspect of the task. Therefore, it is difficult to just select one model as the best. We suggest that future researches of syntactic change detection could take reference from this paper and tailor potential profitable sections into their work.

References

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-bert: Improving pre-trained transformers with syntax trees.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Yanran Chen, Wei Zhao, Anne Breitbarth, Manuel Stoeckel, Alexander Mehler, and Steffen Eger. 2024. Syntactic language change in english and german: Metrics, parsers, and convergences.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jonathan Dunn and Sidney Wong. 2022. Stability of syntactic dialect classification over space and time.

Prof. Edward Flemming. 2019. Language variation and change - section 14: Syntactic change 2. As taught in Spring 2019. Accessed on Date.

L Fonteyn, E Manjavacas, and S Budts. 2022. Exploring morphosyntactic variation and change with distributional semantic models. *Journal of Historical Syntax*, 6:1–41.

Mirjam Fried. 2015. *Construction Grammar*, pages 974–1003.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online. Association for Computational Linguistics.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Oliver Hellwig and Sven Sellmer. 2022. Detecting diachronic syntactic developments in presence of bias terms. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 10–19, Marseille, France. European Language Resources Association.

Liwen Hou and David A. Smith. 2023. Detecting syntactic change with pre-trained transformer models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Renfen Hu, Shen Li, and Shichen Liang. 2019. Diachronic sense modeling with deep contextualized word embeddings: An ecological view. In *Proceedings of the 57th Annual Meeting of the Association for*

Computational Linguistics, pages 3899–3908, Florence, Italy. Association for Computational Linguistics.

Anthony Kroch. 2008. *Syntactic Change*, volume 16, pages 698 – 729.

Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. 2022. Contextualized embeddings for semantic change detection: Lessons learned. *Northern European Journal of Language Technology*, 8(1).

Lei Lei and Ju Wen. 2020. Is dependency distance experiencing a process of minimization? a diachronic study based on the state of the union addresses. *Lingua*, 239:102762.

Xianghe Ma, Michael Strube, and Wei Zhao. 2024. Graph-based clustering for detecting semantic change across time and languages.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014a. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014b. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

William Merrill, Gigi Stark, and Robert Frank. 2019. Detecting syntactic change using a neural part-of-speech tagger. pages 167–174.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*.

Florent Perek. 2016. Using distributional semantics to study syntactic productivity in diachrony: A case study. *Linguistics*, 54(1):149–188.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Ian Roberts and Anna Roussou. 2007. Syntactic change. a minimalist approach to grammaticalization. 129(3):469–476.

Dominik Schlechtweg, Anna Hätty, Marco Del Tredici, and Sabine Schulte im Walde. 2019. A wind of change: Detecting and evaluating lexical semantic change across times and domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 732–746, Florence, Italy. Association for Computational Linguistics.

Zhaochen Su, Zecheng Tang, Xinyan Guan, Juntao Li, Lijun Wu, and Min Zhang. 2022. Improving temporal generalization of pre-trained language models with lexical semantic change.

Ruiyu Wang and Matthew Choi. 2023. Large language models on lexical semantic change detection: An evaluation.

Marcos Zampieri, Shervin Malmasi, and Mark Dras. 2016. Modeling language change in historical corpora: The case of portuguese.