INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Redefining the Linguistic Context Feature Set for HMM and DNN TTS Through Position and Parsing

Rasmus Dall[1], *Kei Hashimoto*[2]
Keiichiro Oura[2], *Yoshihiko Nankaku*[2], *Keiichi Tokuda*[2]

[1]The Centre for Speech Technology Research, The University of Edinburgh, UK
[2]Department of Computer Science and Engineering, Nagoya Institute of Technology, Japan
r.dall@sms.ed.ac.uk, bonanza@sp.nitech.ac.jp
uratec@nitech.ac.jp, nankaku@nitech.ac.jp, tokuda@nitech.ac.jp

Abstract

In this paper we present an investigation of a number of alternative linguistic feature context sets for HMM and DNN text-to-speech synthesis. The representation of positional values is explored through two alternatives to the standard set of absolute values, namely relational and categorical values. In a preference test the categorical representation was found to be preferred for both HMM and DNN synthesis. Subsequently, features based on probabilistic context free grammar and dependency parsing are presented. These features represent the phrase level relations between words in the sentences, and in a preference evaluation it was found that these features all improved upon the base set, with a combination of both parsing methods best overall. As the features primarily affected the F0 prediction, this illustrates the potential of syntactic structure to improve prosody in TTS.

Index Terms: Speech Synthesis, TTS, PCFG, dependency parse, parsing, HMM, DNN, linguistic features

1. Introduction

In Hidden-Markov-Model (HMM) Text-to-Speech synthesis (TTS) a linguistic context feature set is used for decision tree based clustering of the training data. The set of contexts used is derived from linguistic analysis of the text of the training corpus, and contains information about the context of the current phoneme, syllable and word. The standard set of features used in HMM synthesis was proposed together with the first release of the HTS speech synthesis system [1], and notably has not changed since then [2] (see Section 2 for details of this set). This is not to say that researchers have not been interested in this area, however, there is little direct research on the topic.

Research has also shown that, in the current feature set, features above the word level have little to no impact on final speech output quality. In [3] features were slowly removed starting from utterance level features down to syllable level, and it was found that no above word level features had any significant impact on the resulting models. In a similar vein, [4] used a Bayesian network to find the most relevant features from the standard set, and obtained a minimal degradation of output speech quality with a feature set consisting of only 6-9 (depending on stream) features, down from the standard 26. This suggests that many features at the word-level and above do not have a large impact on speech quality. However, it should be possible to find such features.

In particular, the use of parsing derived features has been investigated in a few studies. The system proposed by [5, 6] incorporates parsing derived features, but unfortunately they do not specify which type of parsing, likely Probabilistic Context Free Grammar (PCFG), nor exactly which features were included (Section 3.1.2 in [6] and Section 3 in [5]). The effect of parsing was also only indirectly evaluated using objective measurements in Chinese [5] and by submitting a, very well-performing, English system for the 2013 Blizzard challenge with similar features [6]. It is however unclear whether the system performed well due to the addition of the parsing features or due to other differences, e.g., that it was a hybrid system. In [7] a Finnish system for using parsing derived features for rule-based prominence prediction is described, however they do not put the system to the test. In the Spanish system from [8] (Section 4.3) morphosyntactic features derived from part-of-speech tagging and a parse tree were used. These features improved the synthetic speech quality, but unfortunately which features were used was not detailed. In French, [9] uses the Alpage Linguistic Processing Chain to extract a set of parsing derived features and, in a comparison mean opinion score (CMOS), certain types of sentences show an improvement, while others degrade. Their baseline did, however, include some morphosyntactic features, which are, at least in English, not standard and are arguably parsing derived. Overall this suggests that parsing based features can capture information relevant to TTS systems at the word-level and above, and we therefore present an investigation into using PCFG and dependency parser based features.

Furthermore, deep neural network (DNN) synthesis is becoming increasingly popular and this method of synthesis often combines the decision tree context clustering and acoustic modelling into the neural network (e.g., [10, 11]). Because of this it is likely that the neural network may be able to utilise different features than the HMM system. As such, this work will also focus on some alternative representations of positional values. Positional values are features related to, e.g., phoneme position in syllable or word position in utterance. These values are currently represented as forwards and backwards absolute positional values. Absolute values, however, introduce ambiguity about the meaning of features in segments of differing length, so, in Section 2, we present an investigation into the use of relational or categorical values as an alternative.

We will then present two sets of additional features derived from PCFG and dependency parsing, respectively. Both of these are methods with a long history in Natural Language Processing (NLP), with well defined toolkits (e.g. the Berkeley [12] and Stanford [13] parsers), standard algorithms and active research communities (e.g., [14, 15, 16, 17]). In Section 3, we present each method of parsing and the derived set of additional features

http://dx.doi.org/10.21437/Interspeech.2016-399

followed by an evaluation of the effect of using these features in HMM and DNN synthesis. Section 4 will provide an overall discussion of the findings before concluding in Section 5.

2. Redefining Positional Values

Most available front-ends share a very similar set of linguistic features. Festival [18] and Flite [19] are exactly equivalent, and Idlak [20] uses a subset of the Festival/Flite set, while MaryTTS [21] is overall very similar to Festival/Flite. In this paper we use the Festival/Flite set as our base set, which is also the standard set used in the HTS demo, and it consists of:

- Quinphoneme context and phonetic detail
- Forward/backward position of:
  - Phoneme in syllable
  - Syllable in word, phrase, utterance
  - Word in phrase, utterance
  - Phrase in utterance
- Number of:
  - Phonemes in current/next/previous syllable
  - Syllables in current/next/previous word/phrase
  - Words in current/next/previous phrase
  - Syllables/words in utterance
- General part-of-speech tag of current/next/previous word
- Current syllable accent/stress
- Distance to/from last accented/stressed syllable
- Number of accented/stressed syllables before/after current syllable
- ToBI end-tone marking

As can be seen, many of these values are positional values. These are expressed as the forward and backward absolute position of a segment in a larger segment, e.g., the position of a word in the utterance. So the word "hit" in:

$$The\ man\ hit\ the\ brown\ dog. \quad (1)$$

Has a forward positional value of 3, as it is the third word from the beginning, and a backward position of 4, as it is fourth from the end. There is however an issue with this representation, the values can quickly become seemingly meaningless. If the sentence was "Peter hit the dog" or "the angry man hit the dog" both positional values change without seemingly affecting the word's pronunciation much. This issue is much more pronounced at the beginning or end of a segment as the meaning of, e.g., a forward position of 5 is very dependent on the length of the segment, if the segment is 5 long it is at the end, but if it is 10 long it is in the middle.

We therefore propose two alternative representations which may alleviate this issue. The positional values can be represented as relational or categorical positions. A relational position is the position of the shorter segment in the longer, normalised to a value between 0 and 1 where 0 is the beginning and 1 the end. This removes the need for forward and backward positioning as the relational forward position of 0.1 is always equivalent to the backward of 0.9, reducing the number of features necessary while capturing the same information. But, it does not entirely solve the issue as positional values are now normalised you get small differences between positions with every small change in segment length, e.g., in Sentence (1) the relational position of "hit" would change from 0.4 to 0.5 if it was changed to "the man hit the dog". This necessitates many additional potential values, particularly due to the use of greater/less than questions. This shouldn't be an issue for the DNN system as it uses continuous input values naturally, though a decision tree may have problems modelling this.

Another way of representing the positional values is to use categories. We propose to use the following 4 categories; "beginning" for the first element, "end" for the last, "one" for segments of length one and "middle" for all others. This reduces the number of potential context values to a great extent, however, it results in a loss of granularity. In order to retain some context for segments close to the edges, the category of the previous and following segment was also added.

	Absolute	**Relational**	**Categorical**
Questions	1264	1714	1059
Input Layer	510	503	561
DT Leaves	6015	5999	5937

Table 1: The size of the HMM question set, the DNN input layer and total number of HMM decision tree leaves for different positional representations.

Table 1 shows the size of the question set for the HMM, the size of the input layer for the DNN and the total number of decision tree leaves for all streams. The question set size is after pruning questions not present in the training dataset for each of the three methods. As can be seen, the categorical representation results in a decrease in the number of questions, whereas the relational increases it. For the DNN the size of the input layer increases when using the categorical representation, this is due to the addition of previous and following contexts for edges which necessitates adding additional nodes to the layer.

While the number of questions does not indicate the quality of a feature set, if we look at the size of the learned decision trees (which can be seen as an indication of quality) in Table 1, all three representations result in approximately similar sized trees despite different question set sizes. This suggests that the categorical representation can, with fewer questions, capture the same information as the other two representations, but, by the same token it seems the relational representation needs more questions. However, as all three approaches result in similar sized decision trees this could be an indication that different sets do not result in different synthesis – but this is not necessarily true – so to test this a small listening test was performed.

2.1. Evaluating Positional Values

In order to evaluate the three methods of representation, HMM and DNN voices were trained using each of them. A corpus of 1974 sentences, approximately 2 hours of speech, from a native female British English speaker was used. The HMM systems were built using HTS 2.3 Beta. The DNN systems were built using a system similar to the DNN-MLPG system of [22], but modified to use STRAIGHT, and consisted of a 5-layer feed-forward network. The HMM system employs GV and MLSP postfiltering and the DNN does not, but as we are interested in the effect of the feature types in each system separately, this is no issue, and consequently no comparisons across system type were made.

We compared each of the three methods of representation within each synthesis method (DNN or HMM) in a preference test. 10 native English speakers were recruited, and they performed the test in a sound-proofed booth wearing high-quality headphones. Each participant evaluated each pair of representations, for both HMM and DNN systems, i.e., 6 preference pairs. The same 15 sentences were used for all comparisons. The two

	Absolute	Relational	Categorical
HMM	47.7%	52.3%	-
	-	43.0%	57.0%
	45.6%	-	54.4%
DNN	56.3%	44.7%	-
	-	49.0%	51.0%
	42.0%	-	58.0%
Total	47.7%	47.3%	55.1%

Table 2: Preference scores for each representation pair and the total combined preference for the different positional representations.

sentences within each trial were the same. Sentences were presented in a random order and the order within a trial was randomised to avoid bias effects. In total, each listener gave their preference for 90 trials (15 x 3 HMM and 15 x 3 DNN). The test took approximately 20 minutes to complete. All experimental samples are available here [23].

From Table 2 we can see that, in all cases, the categorical representation is slightly preferred, whereas the relational is preferred over the absolute in the HMM case and the absolute over the relational in the DNN case. These are not statistically significant on their own, partially due to the small test size, but if we pool all datapoints from HMM and DNN (bottom Table 2) we can see that the categorical representation is preferred over the others, and this difference is significant, using the exact binomial test, compared to the absolute and the relational ($p < 0.05$). Therefore, for the rest of this paper, the categorical representation will be used.

3. Parsing Based Features

Two types of parsing were considered. Probabilistic context free grammar (PCFG) and dependency parsing. PCFG parsing is a probabilistic form of syntactic tree-based grammar. A parse tree is created from a set of rules to create a tree expanding from a root node downward from very general phrase types to part-of-speech tags at the leaf nodes. Each rule expansion has a probability assigned to it, and each leaf a probability of expanding into a given word. Finding a parse thus involves finding the most likely parse tree given the words and the possible trees expanding over those words. A PCFG parse thus describes the syntactic phrase structure of a sentence, something which is likely to be helpful for the overall phrase level prosody.

A dependency parse, closely related to shallow semantic parsing, describes the internal relations between words in the sentence. These are relations starting from the root of the sentence, the verb, and relations are then found to the rest of the words in the sentence. The relations describe the internal dependencies between words, such that, e.g., the object of the sentence will have a dependency from it to the verb and so will the subject to the verb and so on, until every word stands in exactly one dependency relation to another. Figure 1 illustrates a PCFG and dependency tree for the sentence "The man hit the dog". Note that not all dependency grammars describe a tree structure with exactly one dependency relation upwards for each word, but for the purposes of this work a purely tree-based representation has been used to derive features.

From both of these parses a number of features were extracted. From the PCFG parse a set of features, likely similar to that of [6], was extracted which were:

- Greatgrand-/grand-/father phrase of the current word
- Position of current/next/previous word in greatgrand-/grand-/father phrase

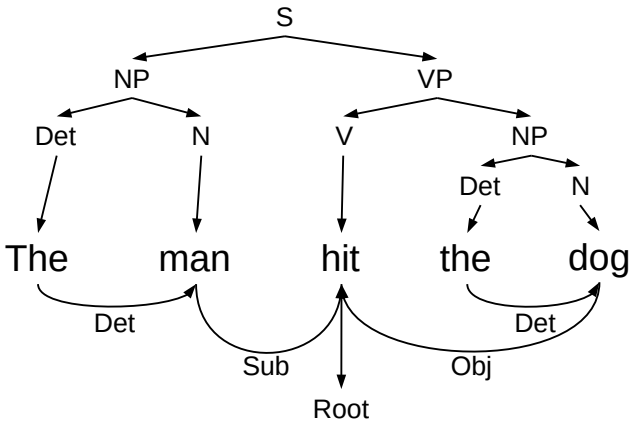

Figure 1: A PCFG (top) and dependency (bottom) parse example.

	Standard	PCFG	Dep	Combined
Questions	1059	1225	1229	1390
Input Layer	561	691	688	817
F0 Leaves	4311	4427	4499	4494

Table 3: The size of the HMM question set, DNN input layer and F0 decision tree leaves for each set of parsing features using the categorical positional representation.

- Expanded general POS-tag

The expanded general POS-tag is an expanded version of the general part-of-speech tag category from Festival, which splits the "content words" in the Festival set to slightly more detailed categories such as "verb"/"noun"/"adjective" but still not the full Penn Treebank [24] set.

From the dependency parse the following features were extracted:

- Current word/father/grandfather to father/grandfather/greatgrandfather relation
- Number of children relations
- Tree arc distance to previous/next word
- Current word distance to father/grandfather/greatgrandfather word
- General relation to father word

The general relation category is defined using the Stanford parser's documentation on the dependency parser (pp. 11-12 [25]) by making most relations one less specific category.

As the features from both types of parsing can be complementary, a Combined set was also created, using all features from both parsing methods. This yielded four different sets of features, a Standard set equivalent to the current Festival/HTS features, a set with the PCFG features added, one with the dependency features added and a Combined set. PCFG and dependency parses were both extracted using the lexicalized parser of the Stanford Parser version 3.5.1.

3.1. Evaluating the Parsing Based Features

For each of the four different sets (Combined, PCFG, Dependency and Standard), an HMM and DNN voice was trained using the same systems as in Section 2.1 and using the categorical positional representation. Each of the 4 voices from each type of system was compared to each other in a preference test, this resulted in 12 (6*2) system pairs. 30 native English speakers were recruited and each participant rated 15 sentences for each pair resulting in a total of 180 comparisons per participant. The

	Standard	PCFG	Dependency	Combined
HMM	48.4%	51.6%	-	-
	47.6%	-	52.4%	-
	43.7%	-	-	56.3%
	-	51.2%	48.8%	-
	-	48.6%	-	51.4%
	-	-	49.8%	50.2%
DNN	42.0%	58.0%	-	-
	45.6%	-	54.4%	-
	34.0%	-	-	66.0%
	-	63.1%	36.9%	-
	-	51.6%	-	49.4%
	-	-	44.8%	55.2%

Table 4: Preference scores for each representation pair and the total combined preference for the differing parsing features.

test had 4 sections of 45 pairs and between each section participants were asked to get out of their booth and walk around a bit before continuing the next section. This was done to avoid listener fatigue. Section order, sentence presentation and pair order was randomised to avoid bias effects. All experimental samples are available here [23].

Table 4 summarises the results. In all cases the parsed versions are preferred over the unparsed. This difference is only significant, using the exact binomial test, when using the Combined set for the HMM ($p < 0.01$), and for the DNN it was significant for the Combined ($p < 0.001$) and PCFG sets ($p < 0.001$) but only marginally for the Dependency set ($p = 0.066$). For the HMM there is very little difference between the three parsed versions, however, for the DNN the Dependency parse is significantly dispreferred (Combined: $p < 0.05$, PCFG: $p < 0.001$) to the other two, between which the difference is not significant ($p = 0.54$). As listeners were asked which of the two sentences in each trial they considered most natural this means that the standard unparsed feature set leads to synthetic speech which is considered less natural than speech generated using the parsed sets, and particularly the DNN system was able to take advantage of these new feature sets.

4. Discussion

The current linguistic context set uses absolute positional values. In our investigation, using categorical positional values provided a better representation resulting in synthesis output which was preferred over synthesis using either absolute or relational values. This difference was more pronounced for the HMM than the DNN system, but the tendency was the same for both. That this had less of an effect in the DNN may be due to the way the categorical feature was represented. For the HMM, using the categorical representation reduced the question set, and thus the possibility for mistakes, by around 20%, presumably one of the reasons for the improvement as the decision tree still captures the salient information using fewer features. However, for the DNN the relational and absolute values were normalised and represented by a continuous value using one input node; but the categorical values were represented using several binary nodes, thus increasing the size of the input layer in a way which may have made it harder for the DNN to deal with. It is also possible that the DNN may simply be able to compensate for the more confusing input of the relational and absolute values in other ways than the decision tree for the HMM.

Deriving features based on PCFG and dependency parsing also improved synthesis. This improvement seems primarily to have come from the prosodic domain, with the F0 decision

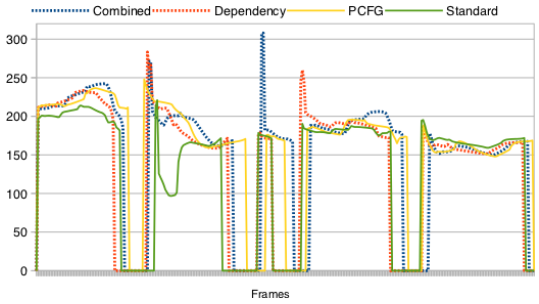

Figure 2: Generated f0 contours for a sample sentence.

tree clustering being the most affected of the different streams (as shown in Table 3). Figure 2 shows a sample sentence and the generated F0 contours, from which we can see that all of the parsing features sets generate a more lively F0 output. This finding makes good sense, the PCFG provides a good representation of the detailed phrase structure of a sentence, improving phrase level F0 movements, and the dependency parse highlights important words in particular relations, improving prominence patterns of a sentence. That these features are complementary is not entirely obvious as the PCFG and Combined sets are equally preferred, it is, however, likely that this may be due to the larger number of features in the Combined set. We have argued that one potential reason why the categorical representation improves over the others is that it describes the same information using fewer features, however, we have then gone on to add many additional features through the parsing derived sets. We have here not attempted to weed out features that are not useful, and consequently the size of our feature set has increased which could lead to confusion in the models. Applying methods for feature set reduction, such as in [4], could help reduce the feature set without affecting the performance, in fact it may help the larger Combined set. However, it does appear that the DNN, which saw the clearest improvement and the largest increase of input size, is capable of dealing with these additional features to some extend. Furthermore, by using the categorical representation, the total number of questions for the Combined set (Table 3) is only a small increase compared to the standard set using absolute positional values (Table 1).

There are many different potential features to derive from parsing, the sets presented here are by no means the only combinations, does not use the only possible parsers and are unlikely to be the best combinations. We encourage others to refine these sets – or find new ones – and to this end we release a research front-end written in Python 2.x which can reproduce all of the presented context sets, and which allows for rapid experimentation with alternative sets. It is available at [26].

5. Conclusions

We have presented an investigation of the linguistic context feature set for HMM and DNN synthesis. Representing positional values using categorical values was found to improve upon the standard absolute and an alternative relational representation. We also found that applying PCFG and dependency parsing can provide additional features, useful for describing the word level interactions, particularly with regard to F0, and that using these features for voice building improves the resulting synthesis.

6. Acknowledgements

This work was supported by the JST CREST uDialogue project.

7. References

[1] H. Zen, "HTS Demo 1.0: Label Overview," 2002. [Online]. Available: http://hts.sp.nitech.ac.jp/archives/1.0/HTS-demo-CMU-Communicator.tar.gz

[2] W. G. HTS, "HTS Demo 2.3: Label Overview," 2015. [Online]. Available: http://hts.sp.nitech.ac.jp/archives/2.3/HTS-demo{_}CMU-ARCTIC-SLT.tar.bz2

[3] O. Watts, J. Yamagishi, and S. King, "The role of higher-level linguistic features in HMM-based speech synthesis," in *Proc. Interspeech*, no. September, 2010, pp. 841–844.

[4] H. Lu and S. King, "Using Bayesian Networks to find relevant context features for HMM-based speech synthesis." in *Proc. Interspeech*, Portland, Oregon, USA, 2012.

[5] Y. Yu, D. Li, and X. Wu, "Prosodic Modeling with Rich Syntactic Context in HMM-Based Mandarin Speeh Synthesis," in *Proc. ChinaSIP*, 2013, pp. 132–136.

[6] Y. Yu, F. Zhu, X. Li, Y. Liu, J. Zou, Y. Yang, G. Yang, Z. Fan, and X. Wu, "Overview of SHRC-Ginkgo speech synthesis system for Blizzard Challenge 2013," in *Blizzard Challenge 2013*, 2013.

[7] A. Suni and M. Vainio, "Deep syntactic analysis and rule based accentuation in text-to-speech synthesis," in *Proc. TSD*, Brno, Czech Republic, 2008, pp. 535–542.

[8] R. Barra-Chicote, J. Yamagishi, J. M. Montero, O. Watts, S. King, and J. Macias-Guarasa, "The GTH-CSTR Entries for the Speech Synthesis Albayzin 2010 Evaluation: HMM-based Speech Synthesis Systems considering morphosyntactic features and Speaker Adaptation Techniques," in *Proc. II Iberian SLTech Workshop*, 2010, pp. 353–358.

[9] N. Obin, P. Lanchantin, M. Avanzi, A. Lacheret-dujour, and X. Rodet, "Toward Improved Hmm-Based Speech Synthesis Using High-Level Syntactical Features," in *Speech Prosody 2010*, 2010, pp. 3–6.

[10] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, Vancouver, Canada, 2013, pp. 7962–7966.

[11] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing Multi-Task Learning and stacked bottleneck features for speech synthesis," in *Proc. ICASSP*, Brisbane, Australia, 2015, pp. 4460–4464.

[12] S. Petrov and D. Klein, "Improved Inferencing for Unlexicalized Parsing," in *Proc. HLT-NAACL*, Rochester, New York, USA, 2007.

[13] D. Klein and C. Manning, "Accurate unlexicalized parsing," in *Proc. ACL*, Sapporo, Japan, 2003, pp. 423–430.

[14] J. Andreas, D. Klein, and C. S. Division, "Alignment-Based Compositional Semantics for Instruction Following," in *Proc. EMNLP*, Lisbon, Portugal, 2015, pp. 1165–1174.

[15] D. Chen and C. D. Manning, "A Fast and Accurate Dependency Parser using Neural Networks," in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 740–750.

[16] G. Durrett and D. Klein, "Neural CRF Parsing," in *Proc. ACL*, Beijing, China, 2015.

[17] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with Compositional Vector Grammars," in *ACL*, Sofia, Bulgaria, 2013.

[18] A. W. Black, P. Taylor, and R. Caley, "Festival 2.4 Documentation," 2014. [Online]. Available: http://www.festvox.org/docs/manual-2.4.0/festival{_}toc.html

[19] A. W. Black and K. A. Lenzo, "Flite: a small fast run-time synthesis engine," in *Proc. SSW*, Perthshire, Scotland, 2001.

[20] M. P. Aylett, R. Dall, A. Ghoshal, G. E. Henter, and T. Merritt, "A Flexible Front-End for HTS," in *Proc. Interspeech*, Singapore, Singapore, 2014.

[21] M. Schröder and J. Trouvain, "The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching," *International Journal of Speech Technology*, vol. 6, pp. 365–377, 2003.

[22] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "The effect of neural networks in statistical parametric speech synthesis," in *Proc. ICASSP*, Brisbane, Australia, 2015, pp. 4455–4459.

[23] R. Dall, "Samples for: Redefining the Linguistic Context Feature Set for HMM and DNN TTS Through Position and Parsing," 2016. [Online]. Available: dall.dk/rasmus/Samples/IS{_}2016/IS2016Samples.zip

[24] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of English: The Penn Treebank." *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[25] M.-C. de Marneffe and C. D. Manning, "Stanford typed dependencies manual," 2015. [Online]. Available: http://nlp.stanford.edu/software/dependencies{_}manual.pdf

[26] R. Dall, "SiRe: (Si)mply a (Re)search Front-end," 2016. [Online]. Available: www.github.com/RasmusD/SiRe