Proceedings of the EURALI Workshop @LREC2020, pages 28–35
Marseille, 20 June 2022

Word Class Based Language Modeling: A Case of Upper Sorbian

Isidor Konrad Maier, Johannes Ferdinand Joachim Kuhn, Frank Duckhorn
Ivan Kraljevski, Daniel Sobe, Matthias Wolff, Constanze Tschöpe
BTU Cottbus-Senftenberg, Chair of Communications Engineering, Cottbus, Germany
{isidorkonrad.maier, johannesfk.kuhn, matthias.wolff}@b-tu.de
Fraunhofer Institute for Ceramic Technologies and Systems IKTS, Dresden, Germany
{ivan.kraljevski, frank.duckhorn, constanze.tschoepe}@ikts.fraunhofer.de
Foundation for the Sorbian People, Bautzen, Germany
daniel.sobe@sorben.com

Abstract

In this paper we show how word class based language modeling can support the integration of a small language in modern applications of speech technology. The methods described in this paper can be applied for any language. We demonstrate the methods on Upper Sorbian.
The word classes model the semantic expressions of numerals, date and time of day. The implementation of the created grammars was realized in the form of finite-state-transducers (FSTs) and minimalists grammars (MGs).
We practically demonstrate the usage of the FSTs in a simple smart-home speech application, that is able to set wake-up alarms and appointments expressed in a variety of spontaneous and natural sentences.
While the created MGs are not integrated in an application for practical use yet, they provide evidence that MGs could potentially work more efficient than FSTs in built-on applications. In particular, MGs can work with a significantly smaller lexicon size, since their more complex structure lets them generate more expressions with less items, while still avoiding wrong expressions.

Keywords: word classes, minimalist grammar, language modeling, speech recognition, Upper Sorbian

1. Introduction

Recently the adoption of speech technologies, particularly speech recognition and dialogue systems has been on the rise. The tech giants (such as Google, Amazon, Microsoft, Baidu) provide speech and voice applications (personal assistants) that support mostly languages with a large population where economic interest exists.

The recent state-of-the-art automatic speech recognition (ASR) systems made a breakthrough in terms of recognition achieving "near-human" performances, however in restricted conditions, domain, and language. Also, the challenges of introducing new languages in state-of-the-art ASR systems are multi-fold, especially if they have limited electronic resources.

It is considered that if enough data for a target language exist or could be collected, then the data amount requirements for reliable speech and language modeling by using end-to-end systems and deep learning would be feasible.

In this study, we present one aspect in the development of speech technologies, namely language modeling for speech recognition in Upper Sorbian (prospectively for Lower Sorbian too) as an example of under-resourced and endangered language.

To overcome the lack of textual data necessary for reliable statistical language modeling, we adopted the word class based approach to model named entities (such as numerals, time, date). They represent reusable language resources that can be combined with both formal grammars and statistical language models in a custom speech applications. The resources including text data, grammar definitions and tools are made publicly available with an open-source license.

1.1. Sorbian Languages

The Sorbian languages are spoken in Lusatia in Eastern Germany. The Sorbian languages consist of Upper- and Lower Sorbian - which have standardized writing systems - and an intermediate dialect continuum. All Sorbian languages except Upper Sorbian are highly endangered by extinction (Moseley, 2012).
They belong to the Western Slavic languages along with Polish, Czech, Slovakian and others. They form a subbranch of the Slavic language family with high degree of mutual intelligibility (Golubović and Gooskens, 2015). Lower- and Upper Sorbian are especially similar to Polish and Czech respectively (Měťšk, 1958).
For detailed linguistic information on Upper Sorbian we recommend Anstatt et al. (2020). Overall, Upper Sorbian is described as a typical Slavic language. Its most notable peculiarities are the dual as a grammatical number and the German influence, especially on vocabulary, sentence structure and pronunciation.

1.2. State-of-the-Art

1.2.1. Speech Technology in Sorbian Languages

Due to the mutual similarity between (West) Slavic languages, cross-dialectal language technology could be employed. This approach has already showed success across Spanish dialects and across Arabic dialects, see (Elfeky et al., 2018).
Note that the division into either different languages or

just different dialects is rarely linguistically but rather politically classified (Weston and Jensen, 2000).
Just as Arabic dialects and Spanish dialects share a common standard language each, also for Slavic languages there is a constructed Interslavic language that is highly intelligible with other Slavic languages (Wierzbicki, 2019).
Specifically, Nědolužko (2019) using Czech language data for Upper Sorbian has been considered.
Sorbian language script has been standardized and integrated into various international norms, like ISO639, BCP47 and Unicode Common Locale Data Repository (CLDR) (Böhmak, 2019). Electronical Lexica for Sorbian words and for Sorbian names have been created, and a text-to-speech function for Lower Sorbian is in development (Bartels et al., 2019).
Based on the lexica an online translator was implemented. It uses a statistical MOSES decoder to translate parts of sentences. A neural system OpenNMT can form grammatically correct sentences out of the parts, see (Brězan et al., 2019). Lately, Microsoft has taken the bilingual speech corpus and added Upper Sorbian support to Bing Microsoft Translator, see (Langkabel, 2022).

1.2.2. Grammar Technology

For modelling grammars, lexical and acoustic models, we use weighted finite-state transducers (FSTs) as well as minimalist grammars (MGs).
FSTs were introduced into speech recognition technology by M. Mohri (1997). They are broadly used in current speech processing toolkits like OpenGrm (Roark et al., 2012) or the Kaldi Speech Recognition Toolkit (Povey et al., 2011). For model size reduction and efficient recognition we use an extension of FSTs for modelling context-free grammars (Duckhorn and Hoffmann, 2012; Allauzen and Riley, 2012).
There has already been detailed work by Torr (2019; Stanojević and Stabler (2018; Fowlie and Koller (2017) in realizing parsers that mimic humans internal parsing with MGs. To increase the performance of the grammars, i.a. Kobele (2018; Kobele (2021) and Ermolaeva (2020) made an effort to make the grammars of MGs more succinct. First steps are already under way, besides this work, to prepare MGs for the use in a natural language processing context (beim Graben et al., 2020; Römer et al., 2022).

1.3. Prior Work

The development of speech technologies in Upper Sorbian, particularly speech recognition, started in 2020 with a feasibility study. It encompassed speech and language resource collection and was successfully concluded in 2021. As a result, valuable resources were provided that can be employed in various speech applications.
In (Kraljevski et al., 2021b) we presented acoustic modeling in the Upper Sorbian language where an acoustic model in German was used in cross-lingual transfer learning. Here, we defined grapheme and phoneme inventories and mapped the Upper Sorbian phonemes to the most similar German equivalents. Then, phonetically balanced sentences were selected from the available textual data (HSB Common Voice project) and combined with application specific ("SmartLamp" use-case) sentences into recording prompts for speech corpus collection.
The original acoustic model in German was utilized to segment and force-align the speech corpus by the knowledge-based phoneme mappings. The quality was evaluated by the resulting confusion matrix of the free phoneme recognition and provided better derived data-driven phoneme mapping. Then, the German acoustic model was acoustically adapted to the recordings in Upper Sorbian and as such implemented in a speech recognition demonstrator for controlling smart home devices ("SmartLamp").
The studio recorded speech corpus comprises of around 11 hours of male, female, and child speakers, with the corresponding metadata, such as text corpus, lexicons, and language models. The collected resources provided the possibility for fundamental research in phonology and phonotactics of Upper Sorbian. Taking advantage of the outcomes of the feasibility study, we conducted a study for a data-driven approach for the quantitative analysis of glottal stops before word-initial vowels in Upper Sorbian (Kraljevski et al., 2021a).
However, the available resources are insufficient to employ state-of-the-art (SotA) speech recognition techniques such as hybrid Hidden-Markov-Model (HMM) combined with a deep neural network (DNN) or even end-to-end DNN, where the inputs are raw and unprocessed utterances and the outputs are the corresponding sequences of graphemes, words or semantic entities.
Therefore, in the follow-up project of the feasibility study concluded in March 2022, we improved the acoustic, lexicon and language modeling, with the aim to further develop the speech recognition in Upper Sorbian, and to extend it for Lower Sorbian.

2. Word Class Language Modeling

Depending on the intended speech application the language model can be defined either by a handcrafted formal grammar or by a statistical language model (SLM). Formal grammars are appropriate for very limited vocabulary (few hundreds to thousand words) where the spoken utterance must follow the expected order of words/morphemes. In contrast, statistical language modeling (SLM) estimates the probability of word sequences based on N-gram statistics (unigrams, bigrams, trigrams, and higher).

To train an SLM, a large amount of text data is required, which in general will never cover all the possible contexts in a given domain and the problem is even more emphasised in the case of under-resourced languages.
For instance, if a textual corpus that contains all the

numbers in the corresponding context is required; it will have a huge size and will be impossible to acquire. Instead, each occurrence of a number in the text is replaced with a label (tag) representing a word class (in this case, numerals). Consequently, training such a word class language model provides significant reduction in the complexity and the vocabulary size. Word class modeling improves the generalization in both statistical- and formal-grammar- based language models.

The concepts are demonstrated in the following sections, where word class modeling is demonstrated on numbers, time and date implemented as weighted finite-state transducer (FST) grammars and minimalist grammars (MG). Since either of the word classes are finite, they can all be generated by regular FST grammars. More powerful grammars - like MGs - can still be used in order to model the word classes with smaller model size.

2.1. Modeling with MGs

We chose to use MGs, because they are considered especially well-suited for modelling natural human language. (Torr, 2019; Stabler, 2013; Fowlie and Koller, 2017; Versley, 2016; Stanojević and Stabler, 2018) In particular, the structural operations enable MGs to draw dependencies between non-adjacent morphemes with little obstacles. However, integration of MGs into State-of-the-Art technologies is mostly still in development. For this reason, we simply present a stand-alone program that can parse a a variety of Upper Sorbian prompt sentences. In order to extend an MG, we add new items that hold a new category and use selectors to connect the new category with the present grammar. Example: In Figure 3 an integer time of day expression gets extend by selecting items of a modifier category and a daytime category.

Exceptions on the possible connections of certain items with categories via selection can be handled with the distribution of licensors and licensees. Example: In Figure 4 the pair $\pm m30$ regulates that 'januar' can form a date by connecting with 'třicety' (30th), while 'februar' cannot.

2.2. Modeling with FSTs

FSTs have a simpler structure compared to MGs and they are a lot more commonly used in State-of-the-Art language technology.

However, the simpler structure implies some limitation for the grammar:

- An FST always needs several transitions for the same morpheme, if the morpheme can appear in different positions in the construction. Hence, FSTs often require a larger model size than MGs do.
- Since dependency relation between morphemes is only controlled by (non-)adjacency of transitions, it is inefficient to model dependencies between non-adjacent morphemes.

  Example: Assume we want to extend a present complex grammar - with start node S and end node E - by the option to put all final expression in brackets. Then all expression have to start with '(', if and only if they end with ')'. The only way to model this dependency is to duplicate the entire complex structure between S and E, which requires to duplicate the entire FST, see Figure 1.

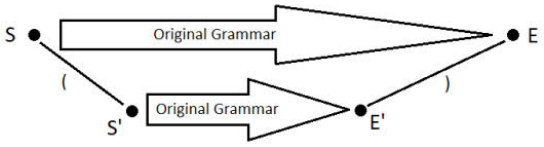

Figure 1: FST model of bracket relation

Sub-grammars can be incorporated by replacing transitions in an FST with another FST. This enables the use for the grammars from the following sub-sections in a larger handcrafted grammar or in a statistical language model.

2.3. Numbers

In either model we first we build a basic grammar of cardinal numerals $1-9$, onto which we construct a grammar for $1-99$. The smaller grammars (in MGs: categories) are used to recursively build larger grammars onto, so we gain grammars for numerals up to 999, then 10^6-1, 10^9-1, $10^{12}-1$ and $10^{15}-1$. In the MG, the rising sets of numbers are represented by distinct categories.

The constructions are not always uniform, so we have to handle exceptions. A frequent exception is that, if the two digits before a decimal power's noun - like "milion", "miliard", "bilion",... - are larger than 4, then the genitive plural "milionow"/"miliardow"/"bilionow"/... is used. On the contrary, $3*10^6$ and $4*10^6$ call the nominative plural ("tři/štyri miliony"), $2*10^6$ the nominative dual ("dwaj milionaj") and $1*10^6$ the nominative singular ("milion").

In the FST model, we handle this by introducing a special subgrammar for numerals $5-99$, while the connection of the numerals $1-4$ with $10^{6/9/12/15}$ is handled individually. The MG model handles it by the distribution of licensors and licensees.

As another exception, there are two words for 50. The expressions "pjećdźesat" (five tens) and "połsta" (half hundred) are arbitrarily interchangeable, even as subexpressions inside other numerals like of 51, 150 or 50000. It is no problem for modelling, but once it comes to generating, a decision making is needed.

For the FST model, we also built some special numeral grammars like NUM0-23 and NUM0-59, which

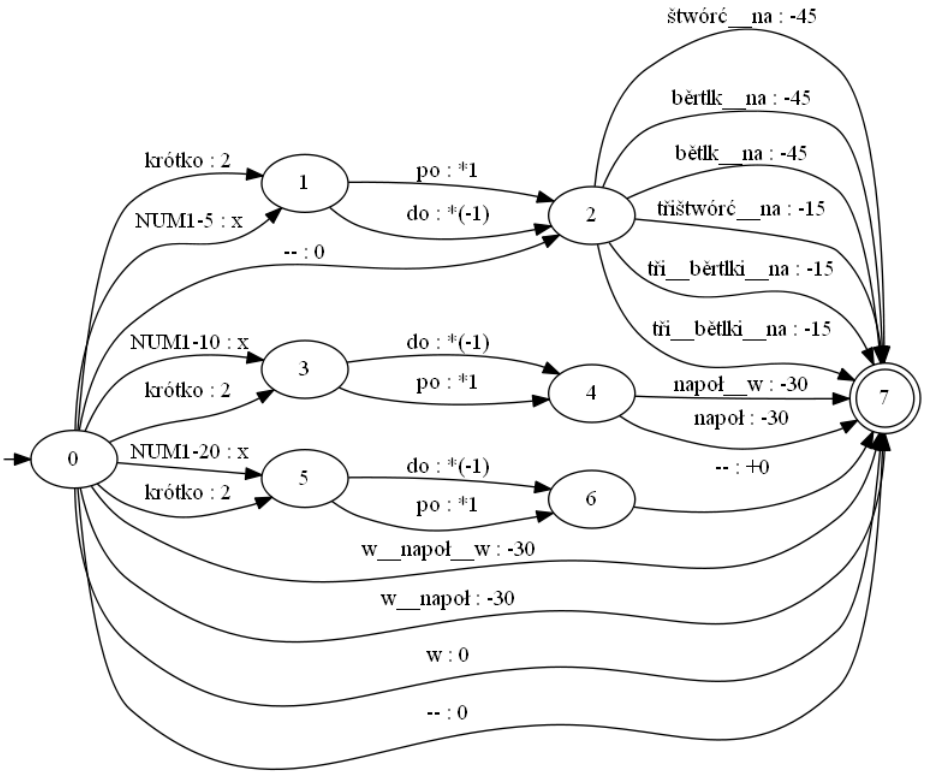

Figure 2: Sorbian hour time modifiers.
Note: NUMz-y represents a subgrammar FST for the numerals from z to y.

become incorporated in the time of day FST grammar as hour and minute counts. We also built ordinal numeral FST grammars ORD1-29, ORD1-30, ORD1-31 (all masculine gender) and a grammar for feminine ordinals ORD1-31f, which become incorporated in the date FST grammars as day counts of the months.
In the MG model, these special grammars are directly built into the time of day MG and date MG respectively.

2.4. Time of Day

The time of day grammars convert time of day expressions into a numerical representation of the time. As a representation we do not use the classical $hh : mm$ format but rather just the count of minutes after midnight. We assume that this one-dimensional representation is not just easier to compute, but also easier to work with in the post-process. For the purpose of a printout, a minute count m can easily be converted into $hh : mm$ with:

$$hh = \lfloor \frac{m}{60} \rfloor \; mod \; 24 \text{ and } mm = m \; mod \; 60. \qquad (1)$$

Note that the minute count output does not necessarily need to be between 0 and 1440, but may also be negative. Still, the mod-operator in (1) will always lead to an hour computation between $0 - 23$.
We are considering two different types of time expressions.

- One covers the accurate digital expressions, that are mostly used in official exact speech and for odd appointments of, e.g. a bus or train departure. These expressions can be simply modeled out of a numeral between $0 - 23$ as the hour count, a numeral between $0 - 59$ as the minute count and "hodźin" as a connection word between the hour and minute count.

- The second type covers the more common - but complicated - everyday expressions like "tři štwórć na pjećich" (corresponding to "quarter to five"). We modeled this type of expressions as a construction out of 3 blocks.

  – The first block can represent the daytime (morning, noon,...). It is important to include into the grammar, since it does influence the meaning. For instance, "six in the morning" has a different meaning than "six in the evening".

  – The second block consists of any modifier of the clocktime, like in English "X/quarter to/past" or "half past" or even combinations out of those. We discussed with the client - the Foundation for the Sorbian People - in order to agree on which combinations of different modifiers should be covered. We agreed on the sub-grammar presented in Figure 2

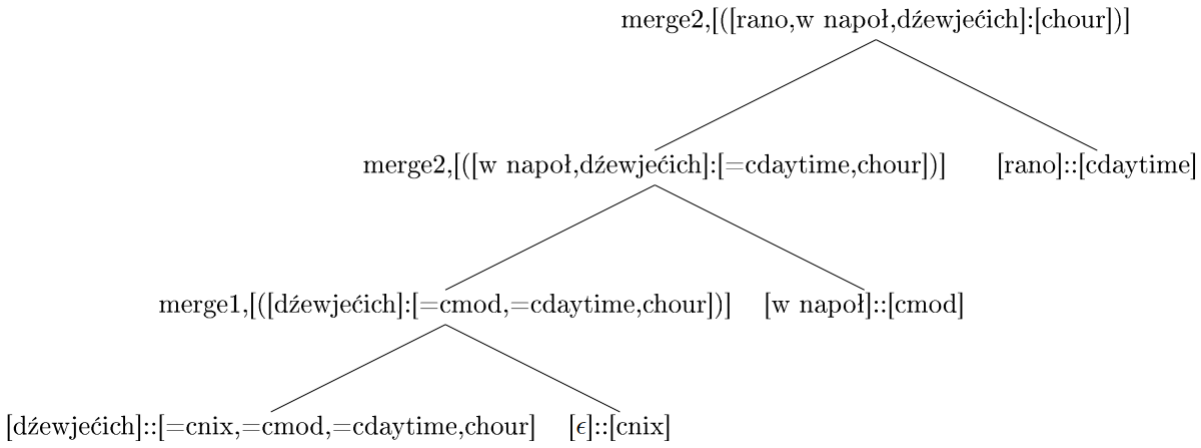

Figure 3: The integer time 'dźesaćich' (9) is extended by a modifier 'w napoł' (half to) and a daytime specifier 'rano' (morning).
'dźesaćich' first merges with a neutral item, so it becomes a derived item. As a derived item, it places the further items it merges with to the left side (Merge2). So, the last merged 'rano' ends up non-adjacent to 'dźewjećich', despite being selected by it. This way, the MG can handle non-adjacent dependencies.

– The third part numerically represents the related hour. In Upper Sorbian everyday speech a 12 hour format is used, so e.g. "dwěmaj" could either mean 2:00 or 14:00. The actual meaning can be determined based on the daytime.

The blocks of daytime and hour have a dependency. For instance, 'six' cannot be connected with 'noon'. If it could, it would even be highly ambiguous, whether it means 6 or 18.
This dependency of non-adjacent parts of the expression makes the FST model laborious. As shown in Figure 1, it requires to include several copies of the modifier grammar (Figure 2) - one for each day time - into the grammar.
An MG can handle non-adjacent dependencies as shown in Figure 3.

2.5. Dates

For the date grammars, we again decided to represent the numerical meaning in a single number - the day count after New Year's Eve. Again, we will also - and even mostly - use negative numbers. Since there are leap years, the day count after New Year's Eve is indefinite for any date from March to December. However, the day count till New Year's Eve is definite, so we use negative counts for March till December but positive counts for January and February. So, the output is always a number between -305 and 60.
Moreover, we built two different date grammars for nominative and genitive case. Both cases are needed for some significant wordings.
The date grammars are built out of an ordinal number representing the day and a name for the month. We included 3 different names for each month: A numerical name as the ordinal number of the month, a Gregorian name and an older traditional name.

In the FST model we combine the month names with the ordinal number grammars ORD1-29, ORD1-30, ORD1-31 or ORD1-31f, see 2.3, depending on the length and gender of the month (name).
In the MG model we instead handle the combinations with the license pairs as shown in Figure 4.

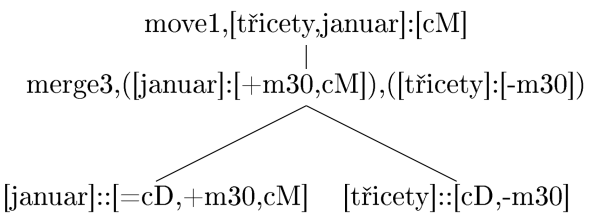

Figure 4: The licensor $+m30$ allows 'januar' to become a terminating date expression after merging with 'třicety' (30th).
'februar' would not hold this licensor. So, if 'februar' merged with 'třicety', the construction could not terminate, since 'třicety's licensee $-m30$ would never be triggered.

3. Practical Implementation

3.1. Finite-State-Transducers (FSTs)

We implemented the grammars to represent different building blocks for the numerals, time, and date. We created tools that combine FST based language models (handcrafted grammars or statistical language models) containing word class tags with the corresponding word class grammars. The resulting language model in OpenFST (Allauzen et al., 2007) format and the corresponding lexicon are included in the configuration for the dLabPro/UASR speech recognizer (Hoffmann et al., 2007). Since the OpenFST format is interchangeable, it could be easily incorporated within other speech recognition frameworks.

We developed word class grammars and combined them together with lexicon and phoneme models into an FST, which translates directly from speech frames to semantic token sequences.

During the recognition, the decoder searches all allowable sequences of tokens to find the one that matches the speech the best acoustically.

We evaluated the functionality of the grammars by speech recognition experiments on a small set of fifteen audio examples recorded by one speaker.

The output of the recognizer was analysed in terms of word error rates (WER) and character error rates (CER) for the semantic concepts and used to debug the grammars. The following example ("Make appointment Wednesday evening at seven.") shows such a recognition result:

```
Ref-Words: ČIN TERMIN SRJEDU WJEČOR W SEDMICH
Res-Words: ČIN TERMIN SRJEDU WJEČOR SEDMICH
Word-ER 16.7% C=5 I=0 D=1 S=0

Ref-Sematic: ČIN TERMIN
<WDAY>+3</WDAY><TIME>+720+0+420</TIME>
Res-Sematic: ČIN TERMIN
<WDAY>+3</WDAY><TIME>+720+0+420</TIME>
Char-ER 0.0% C=48 I=0 D=0 S=0
```

The "Ref-Words" denotes the reference transliterations, while "Res-Words" the results of the speech recognition. The error rates are calculated from the number of correctly recognized tokens (C), insertions (I), deletions (D) and substitutions (S). Similarly, the "Ref-Semantic" and the "Res-Semantic" denote the reference and the recognized semantic expressions respectively, with the corresponding tags (<WDAY>- weekday, <TIME>- time of day). The expressions were calculated as described in the Sections 2.4 and 2.5.

The semantic meaning in three of fifteen sentences were wrongly recognized, mostly due to the recognition errors because of the simple acoustic modeling and missing pronunciation variants. The errors are mostly wrongly recognized months: "WOSMEHO JUNIJA" as "WOSMO JULIJA" ("eight of Juni" as "eight of July"), "SEDMEHO SEPTEMBRA" as "SEDMEHO SEDMO" ("seventh of September" as "seventh of July").

The software tools, the guidelines and all the needed resources are published in a repository of the Foundation for the Sorbian People [1] under the MIT license.

The repository contains fully reproducible examples of both approaches (CFG and SLM) using word classes for language modelling. The resources can be used for building custom word class grammars to be used in practical and more complex speech applications, such as personal voice assistants, meeting protocol transcriptions and dictation of domain specific texts (such as in law, medicine, industry).

[1] https://github.com/ZalozbaDev/speech_recognition_language_modeling

3.2. Minimalist Grammars (MGs)

Regarding the MGs, since its integration into speech technology is still in development, their practical use nowadays is rather limited. We restrict ourselves to parsing the mentioned "Res-Words" from a written form.

Since we have no tools to search through the sentences after outputs of a minimalist (time of day/date) grammar, we build an extra MG of prompts that builds the sentences with variable inputs of times of day and dates.

Another issue for our parser are the numerous ϵ-items - items with empty phonetic part. From a phonetic point of view, any amount of them could be anywhere in the sentence. So, they create a need for greater look ahead in the structural part of the grammar than the currently used parser can manage in a reasonable amount of time. To overcome the problem, we gave all ϵ-items a virtual phonetic 'e', which was also built in the sentences at all places they virtually appear at. Still, even then our parser had to find out, which combination of the 69 ϵ-items in the grammar is needed. So, it still required hours to parse a single sentence.

After we numerated the virtual phonetics by calling them 'e1',...,'e69', the sentences could be parsed in real time.

4. Conclusions and Future Work

We have presented word class based language modeling applied in the case of Upper Sorbian. The word classes model the semantic expressions of numerals, date and time of day. The implementation of the created grammars is realized in the form of finite-state-transducers (FSTs) and minimalists grammars (MGs). The latter realization is a novelty in speech technology.

The usage of the FSTs was practically demonstrated in a use-case of a simple smart-home speech application in Upper Sorbian. It is able to set wake-up alarms and appointments with numerals, date and time of day expressed in a spontaneous and natural speech.

In order to make the speech application more widely usable, more example prompts can be added and the speech recognizer can be trained by more different speakers to improve it.

The created speech and language resources are publicly available as open-source and can be used as building blocks to develop more complex speech applications.

Our future work will be focused on developing an MG framework that is more flexible and user friendly in development and computationally more efficient in practical deployment.

We expect, that our MG parser will soon be able to generate the semantic of parsed sentences, so it could work as a translator of natural prompts into machine readable lambda expressions. An optimization of the detection of ϵ-items would greatly improve the

efficiency and applicability of the program.

Generally, future applications will not be restricted to speech recognition, and in only one language, but also applicable in a wide range of language independent Natural Language Processing (NLP) applications.

5. Bibliographical References

Allauzen, C. and Riley, M. (2012). A pushdown transducer extension for the openfst library. In Nelma Moreira et al., editors, *Implementation and Application of Automata*, volume 7381 of *Lecture Notes in Computer Science*, pages 66–77. Springer, Berlin Heidelberg.

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., and Mohri, M. (2007). OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23. Springer. `http://www.openfst.org`.

Anstatt, T., Clasmeier, C., and Wölke, S. (2020). *Obersorbisch. Aus der Perspektive der slavischen Interkomprehension*. Narr Francke Attempto Verlag, Tübingen.

Bartels, H., Wölke, S., Szczepańska, J., and Měškank, J. (2019). Digitalne rěcne resurse: pśeglěd - doglěd - póglěd. Presented at the "Konferenz zum Thema "Digitalstrategie & sorbische Sprache", Bautzen".

beim Graben, P., Römer, R., Meyer, W., Huber, M., and Wolff, M. (2020). Reinforcement learning of minimalist grammars. *CoRR*, abs/2005.00359.

Brězan, B., Wenk, J., and Langner, O. (2019). Online-Übersetzer deutsch-sorbisch und sorbisch-deutsch - herausforderungen und lösungen. Presented at the Konferenz zum Thema "Digitalstrategie & sorbische Sprache", Bautzen.

Böhmak, W. (2019). Unicode common locale data repository und standardisierung -für die sichtbarkeit der sorbischen sprache in der digitalen welt. Presented at the "Konferenz zum Thema "Digitalstrategie & sorbische Sprache", Bautzen".

Duckhorn, F. and Hoffmann, R. (2012). Using context-free grammars for embedded speech recognition with weighted finite-state transducers. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association (Interspeech 2012)*, pages 1003–1006, Portland, OR, USA, September.

Elfeky, M. G., Moreno, P., and Soto, V. (2018). Multi-dialectical languages effect on speech recognition: Too much choice can hurt. *Procedia Computer Science*, 128:1–8. 1st International Conference on Natural Language and Speech Processing.

Ermolaeva, M. (2020). Induction of minimalist grammars over morphemes. *Proceedings of the Society for Computation in Linguistics*, 3(1):484–487.

Fowlie, M. and Koller, A. (2017). Parsing minimalist languages with interpreted regular tree grammars. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 11–20.

Golubović, J. and Gooskens, C. (2015). Mutual intelligibility between west and south slavic languages. *Russian Linguistics*.

Hoffmann, R., Eichner, M., and Wolff, M. (2007). Analysis of verbal and nonverbal acoustic signals with the Dresden UASR system. In Anna Esposito, et al., editors, *International Workshop on Verbal and Nonverbal Communication Behaviours. COST Action 2102*, volume 4775 of *Lecture Notes in Computer Science*, pages 200–218, Vietri sul Mare, Italy, March. Springer-Verlag. ISBN 978-3-540-76441-0.

Kobele, G. M. (2018). Lexical decomposition. *Computational Syntax lecture notes*.

Kobele, G. (2021). Minimalist grammars and decomposition. *The Cambridge Handbook of Minimalism. Cambridge University Press, Cambridge, to appear*.

Kraljevski, I., Bissiri, M. P., Duckhorn, F., Tschoepe, C., and Wolff, M. (2021a). Glottal Stops in Upper Sorbian: A Data-Driven Approach. In *Proc. Interspeech 2021*, pages 1001–1005.

Kraljevski, I., Rjelka, M., Duckhorn, F., Tschöpe, C., and Wolff, M. (2021b). Cross-lingual acoustic modeling in upper sorbian – preliminary study. In Stefan Hillmann, et al., editors, *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2021*, pages 43–50. TUDpress, Dresden.

Langkabel, T. (2022). Microsoft nimmt obersorbisch in den bing-translator auf. Microsoft News Center.

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311.

Moseley, C. (2012). *The UNESCO atlas of the world's languages in danger: Context and process*. World Oral Literature Project.

Mětšk, F. (1958). Serbsko-pólska rěčna hranica w 16. a 17. lětstotku. In *Lětopis, Reihe B, Band III*, pages 4–25, Budyšin [Bautzen]. Ludowe nakładnistwo Domowina.

Nědolužko, A. (2019). Maschinelles erkennen der tschechischen sprache. vorsprung für die sorben? Presented at the "Konferenz zum Thema "Digitalstrategie & sorbische Sprache", Bautzen".

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December. IEEE Catalog No.: CFP11SRW-USB.

Roark, B., Sproat, R., Allauzen, C., Riley, M., Sorensen, J., and Tai, T. (2012). The opengrm

open-source finite-state grammar software libraries. In *ACL 2012 System Demonstrations*, pages 61–66. http://www.opengrm.org.

Römer, R., beim Graben, P., Huber-Liebl, M., and Wolff, M. (2022). Unifying physical interaction, linguistic communication, and language acquisition of cognitive agents by minimalist grammars. *Frontiers in Computer Science*, 4.

Stabler, E. (2013). Two models of minimalist, incremental syntactic analysis. *Topics in cognitive science*, 5, 06.

Stanojević, M. and Stabler, E. (2018). A sound and complete left-corner parsing for minimalist grammars. In *Proceedings of the Eight Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 65–74.

Torr, P. (2019). *Wide-coverage statistical parsing with minimalist grammars*. Ph.D. thesis, 10.

Versley, Y. (2016). Discontinuity (re) ^{2}-visited: A minimalist approach to pseudoprojective constituent parsing. In *Proceedings of the Workshop on Discontinuous Structures in Natural Language Processing*, pages 58–69.

Weston, T. B. and Jensen, L. M., (2000). *Cultural and regional diversity*. Lanham, Md, Rowman & Littlefield Publishers.

Wierzbicki, N. (2019). Interslavic language — will bulgarian, polish and croatian understand a constructed language? — #1.