Bachelor's Thesis
presented to the Faculty of Arts and Social Sciences
of the University of Zürich
for the degree **Bachelor of Arts**

Computer-Assisted Diagnostics of Developmental Language Disorder in German-Speaking Children

Author: Sofia Carpentieri
Matriculation Number: 19-743-624

Supervisor: Dr. Sarah Ebling

Department of Computational Linguistics

Submission: 1.6.2022

Abstract

This paper presents an extension of a pipeline by Conrad [2021] to help detect Developmental Language Disorder in children's speech. The implementation focuses on two signs for Developmental Language Disorder: wrong subject-verb agreement and incorrect formation of plurals. I created two functions which are included in the pipeline. Both return the output in the form of a TSV file, which can be used for further analysis or training a machine learning model. An evaluation of the automatic annotation of 74 sentences with 145 possible errors yields an observed agreement of 78%. Cohen's Kappa was calculated to compare the inter-annotator agreement and scored 0.275. Finally, the paper offers an outlook and concrete ideas for further work.

Zusammenfassung

Diese Arbeit präsentiert eine Ausweitung der Pipeline von Conrad [2021], die der Erkennung von Sprachentwicklungsstörungen in Sprache von Kindern helfen soll. Die Implementation ist auf zwei Anzeichen für eine Sprachentwicklungsstörung fokussiert: falsche Subjeke-Verb-Kongruenz und inkorrekte Pluralbildung. Ich habe zwei Funktionen geschrieben, die in der Pipeline eingebaut sind. Beide geben den Output in Form einer TSV-Datei zurück, welche für weitere Analysen oder Training von Machine-Learning-Modellen benutzt werden kann. Eine Evaluation der automatischen Annotation über 74 Sätze mit 145 möglichen Fehlern, wurde eine Übereinstimmung von 78% erreicht. Um das Inter-Annotator-Agreement zu bestimmen, wurde Cohen's Kappa berechnet. Ein Ergebnis von 0.275 wurde erreicht. Die Arbeit bietet ausserdem einen Ausblick und konkrete Ideen für zukünftige Arbeit.

Acknowledgement

My greatest thanks go to my supervisor Dr. Sarah Ebling. This thesis would not have been possible without her help, guidance and inspiring suggestions. I would also like to thank Susanne Kempe for providing me with children's speech recordings for my evaluation. Special thanks go to Jessica Roady for her proofreading and helpful suggestions regarding both language and content. Thanks also go to Eyal Dolev for taking the time to read my thesis and giving me valuable advice. I would also like to acknowledge the help of Julia Bischof, who gave me insight into the psychological aspect of children's language disorders and literature in the field. Lastly, I would like to thank my family and friends for supporting me through out my work, even if they did not always understand what I was talking about.

Contents

List of Figures

List of Tables

List of Acronyms

DLD	Developmental Language Disorder
IAA	Inter-annotator agreement
LD	Language disorder
LI	Language impairment
ML	Machine learning
NLP	Natural language processing
POS	Part-of-speech
SALT	Systematic analysis of language transcripts
SLI	Specific Language Impairment
TD	Typically-developing
TSV	Tab-separated values
WHO	World Health Organisation

1 Introduction

Developmental Language Disorder is frequently underdiagnosed and can incur serious, long-term effects on patients' social lives and educational achievements (Grimm et al. [2004], Clegg et al. [2005]). Many who have been diagnosed with Developmental Language Disorder as a child never learn their mother tongue to its full extent, and those who do fall behind their typically-developing peers. Social isolation and insufficient school performance lead to a vicious cycle of language-learning difficulties. Early diagnosis can prevent such consequences, and children as young as 12 months can present the early signs of Developmental Language Disorder. A lengthy clinical observation period is necessary for diagnosis, however, which makes it difficult to screen many children in a short amount of time.

One goal of language disorder research is a machine learning model which can make a diagnosis with high accuracy. It is much more efficient than human labour, which allows more children to be screened. Such a model requires large amounts of training data. Optimally, the data is a collection of children's speech samples annotated with information about the child's language development. Such data, however, is scarce - not only are children's speech recordings much fewer in number than adult's speech, but such specific data as a model needs (for example, speech recordings of German-speaking children with Developmental Language Disorder) is hard to find. The pipeline extended as part of my Bachelor's thesis is a first approach for generating this data, while simultaneously helping therapists to screen many children for Developmental Language Disorders. Its usage does not require any programming skills, making it accessible to therapists and clinicians.

The pipeline receives a voice recording, transcribes it, and analyses the transcription. The analysis can be done on many different aspects of speech. I implemented an initial step for morphological analysis. Since incorrect subject-verb agreement and wrong formation of plurals are indicators for Developmental Language Disorder (de Langen-Müller et al. [2012], Kauschke), I extended the pipeline to make this analysis. The pipeline creates some of the data necessary for training an eventual machine learning model for the automatic diagnosis of Developmental Language Disorder in children.

My thesis is split into three main chapters: *Developmental Language Disorder and Natural Language Processing*, *The Annotation Pipeline* and *Conclusion and Future Work*. The first, Chapter 2, introduces Developmental Language Disorder and an overview of current approaches in the field of natural language processing. I clarify recent terminological changes, then focus on signs that might indicate Developmental Language Disorder. Although such signs can appear on any linguistic level, I focus here on that of syntax and morphology. Further, I present recent natural language processing approaches in the field. I conclude with a presentation of the pipeline on which I base my work created by Conrad [2021].

Chapter 3 is a guide to my implementation. I explain the workings of my functions in the code and provide results and examples of the output. Additionally, I completed an evaluation on a sample audio file. The results of the evaluation are discussed in the last part of the chapter.

In Chapter 4 I conclude my thesis and provide an outlook for further work. The chapter includes concrete suggestions for further implementations of the pipeline.

2 Developmental Language Disorder and Natural Language Processing

2.1 Overview

Developmental Language Disorder (DLD) is a language disorder which affects both speech production and speech understanding, but is not linked to any biomedical conditions. According to de Langen-Müller et al. [2012], 6–15% of German-speaking children are affected by DLD. Grimm et al. [2004] evaluated the speech of monolingual German-speaking children and found that 9.7% of them had DLD, in addition to 19.8% of children suspected to have one. This large gap demonstrates the typical underdiagnosis of DLD that can cause long-term effects on children. Not only are they more likely to encounter obstacles in school and academics, but their social lives can also be affected (Clegg et al. [2005]). Generally speaking, children with DLD only begin learning their mother tongue by the time their typically-developing (TD) peers already communicate very well resulting in the social isolation of children with DLD. This isolation consequently makes it harder for children to learn the language from and alongside their TD peers. This cycle further negatively impacts the children's linguistic and social development. An early diagnosis is thus key to preventing long-term effects in children with DLD.

2.2 Terminology

Previously, DLD has been referred to as 'Specific Language Impairment' (SLI). SLI is a common term in older literature whose use is no longer recommended. Bishop et al. [2017] gathered the CATALISE consortium of 57 professionals in the fields of speech-language therapy, psychology, education, paediatrics, and audiology to discuss the terminology surroundig this condition. In the CATALISE paper, they write that the term 'Specific Language Impairment' has become controversial, as it does not reflect various clinical realities and excludes children from services. In contrast to 'impairment', the term 'disorder' places greater emphasis on the seriousness and importance of the condition associated. The consortium concluded that the word 'specific' has confusing and misleading connotations and proposed to abolish it. On the other hand, 'disorder' might be understood as a problem from 'within the child', although it can change depending on the child's surroundings. Another suggestion, the term 'language delay', was widely rejected. Eventually, the consortium decided to retain the term 'disorder'. In this paper, I will refer to the condition as 'Developmental Language Disorder', even if it is called 'Specific Language Impairment' in previous work, to reflect current terminology and to promote inclusivity.

2.3 Developmental Language Disorder

Language disorders (LD) can be split into many different types of disorders, depending on the child's environment and physical characteristics. Environmental factors include a child's surroundings such as where they live, their parents' socio-economic status, and their friendships. Physical characteristics refer to those parts of the body relevant for speech and language processing, such as the brain and the vocal cords. Language disorders due to atypical brain function or disrupted vocal cords are to be distinguished from disorders not connected to any physical impairments. Diagnoses distinguish between DLD, DLD with comorbidity, and disorders independent from DLD, such as aphasia or disruption of oral fluency. DLD in connection with comorbidity (sometimes referred to as unspecific language impairment, or ULI) includes DLD which is, among others, caused by low intelligence, sensory impairment, or an Autism Spectrum Disorder. Any of the above conditions on their own do not automatically point to a DLD. For my thesis, I will focus on the first form mentioned – DLD. It is estimated that 5–8% of German-speaking children have a DLD, three in four being boys, as Suchodoletz [2003] estimates. Since the term 'SLI' is no longer recommended, the World Health Organisation (WHO) no longer distinguishes between expressive and receptive SLI. In the 11th version of their International Classification of Diseases, the WHO explicitly mentions that a DLD is to be distinguished from an Autism Spectrum Disorder, diseases of the nervous system, deafness not otherwise specified, and selective mutism.

Signs for a DLD can occur very early on. Up to the age of 24 months, starting to speak very late (*late talker*) or not at all is an indicator of a DLD. If children do not match the speech development of their TD peers by the age of 36 months, they should be considered for a diagnosis. Even if children show no significant delay in speech development, there are further signs which indicate a DLD. These are typically split into four levels: (de Langen-Müller et al. [2012], Kauschke)

- **Lexicon and semantics**
  Symptoms on the lexical and semantic level include a very small vocabulary, slow learning of new words, issues with word finding, naming errors, use of so-called *passe-par-tout*-words ('this *thing*'), and unspecific *yes/don't-know* answers.

- **Syntax and morphology**
  Syntax and morphology pertain to the grammar of a child. Usually, children with DLD have issues with case markers, subject-verb agreement, and syntax. They may also struggle to understand complex sentences and questions.

- **Phonology**
  Signs in phonology include the omission and confusion of phonemes, reduced phonemes, and overall issues in pronunciation. Note that impairments due to phonetic-articulatory disorders are excluded.

- **Pragmatics**
  Lastly, issues with pragmatics refer to the reduced ability to understand and use language according to specific situations. Children struggle with turn- taking in dialogue, misunderstand speech acts, and have trouble using nonverbal communication and organising their narrations.

To diagnose a DLD, a child needs to be monitored carefully over long periods of time. This is very costly in both money and time, and there are by far not enough specialists in the field to monitor every child. A precise and effective automated (pre-)diagnosis would thus significantly help to mitigate the effects of underdiagnosis. A pre-diagnosis is – unlike a diagnosis – not medical and only suggests, not confirms, the presence of a disorder. Diagnosis can be made on the basis of both inclusionary and exclusionary criteria. For example, a child with poor language skills that cannot be attributed to neurological deficits, hearing problems, oral-motor skills, or lower than normal-range intelligence will be diagnosed with a language impairment (LI), Solorio [2013] writes. They also note, however, that tests to diagnose LI are clinician-dependent. Although many tests have been developed, results are not always accurate. For some tests, children with learning disabilities will also perform poorly. The tests are also norm-based, which means that they are biased against a poorly represented population. Many factors, such as socio-economic status and ethnic background, can impact the outcome of a test. For example, children from low-income families tend to have a smaller vocabulary than their peers from high-income families (Fernald et al. [2013]), which does not necessarily mean that they have a LI. Furthermore, the tests are often conducted in an artificial environment and can yield results very different from the child's daily communication and interaction. Analysing spontaneous language samples has thus become the focus of attention in recent years. Spontaneous speech can be examined on many different levels, such as grammar, vocabulary, oral fluency, and narrative performance, and can create a more natural environment for the child.

In my work, I focus on the level of syntax and morphology and the aspects of incorrect plurals and subject-verb agreement. Using natural language processing (NLP) techniques, these aspects can be extracted from transcriptions of children's spontaneous speech.

2.4 NLP Approaches

Automating the process for a (pre-)diagnosis enables specialists to screen many children in a shorter period of time than with individual and manual screening. A machine learning (ML) model that can accurately classify children's speech requires large amounts of training data. Unfortunately, children's speech data – especially that which is optimised for machines and ML – is scarce. Thus, it is first necessary to *create* data with which to eventually train the ML model. As Grill and Tuckova [2016] demonstrate in their study, results look promising once such data is available.

Grill and Tuckova [2016] created and analysed speech databases of TD children and children with a DLD. They found that many different classification methods can be applied to children's speech samples. The recordings are well distinguishable, meaning they can be used as data to train a ML model. A ML model learns using different features of the training data. If more distinguishable features are available, classification is easier for the model and can be more accurate and fine-grained. In the software documentation for Systematic Analysis of Language Transcripts (SALT), Miller et al. [2019] also describe different ways to analyse and interpret language samples. Using various types of databases, the authors find numerous measures and features to identify disordered speech. In ML, the availability of many different features is optimal. The more features a machine can learn and weight in its training, the more accurate the results will be.

The work on which I base my thesis is a pipeline for automatic transcription and manual annotation of German speech data by Conrad [2021] (see Figure 1). As a first step towards the automatic diagnosis of conditions involving speech impairment, Conrad [2021] implemented several steps that ease the diagnostic process.

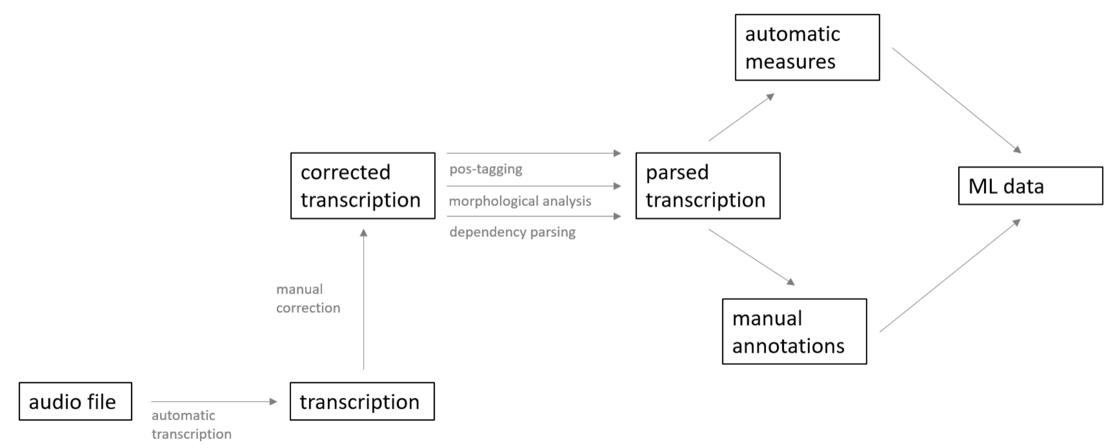

Figure 1: Overview of the pipeline by Conrad [2021].

As input for the pipeline, the user provides a speech recording which is then transcribed. The automatic transcription is done via IMS-Speech by the University of Stuttgart[1]. IMS-Speech is a web–based tool for automatic transcription of audio files created by Denisov and Vu [2019]. Their model has been developed for the German language and outperforms other state-of-the-art systems in automatic speech recognition. The user can manually correct any errors. To do so, the customisable annotation tool Prodigy allows the user to edit the transcription provided by IMS-Speech while simultaneously listening to the original audio file. The user can also tag the corrected transcription using individually-created labels. Once the corrected transcription is finished, the automatic analysis process can begin. The automatic analysis proposed by Conrad [2021] includes information about lexical diversity and lexical density. The diversity of a text is calculated using different measures, such as type-token ratio, moving-average type-token ratio, Brunét's index and Honoré's statistic. The lexical density is the proportion of content words (nouns, verbs, adjectives, and adverbs) with respect to all tokens. My own implementation will extend the functionalities of the automatic morphological analysis in the pipeline.

[1] https://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/ims-speech/

3 The Annotation Pipeline

3.1 Extension Of the Pipeline

I implemented an additional four functions to the pipeline created by Conrad [2021]. After adapting the existing code to run with updated dependencies, I created a file `morphological_analysis.py` in which I wrote four functions. The full code can be found on GitLab.

- `subject_verb_agreement()`
  This function extracts subject-verb tuples and checks for matching grammatical person and number.
- `wrong_plurals()`
  This function searches all nouns in a text and checks whether their plurals have been formed properly.
- `list_to_tsv()`
  As the output of the mentioned functions are lists, these must be converted to strings to write to a tab-separated values (TSV) file.
- `run_analysis()`
  Lastly, this function allows all the code to be run via a single function call.

To implement the automatic morphological analysis, I focused on two main signs for a possible DLD: problems with subject-verb agreement and the formation of plurals. For each of the analyses, I wrote a function which is called in the main function of the program, `run_analysis()`. The user has the option to choose the type of analysis and whether the output should be stored in a TSV file or simply printed to the shell. Figure 2 illustrates the pipeline.

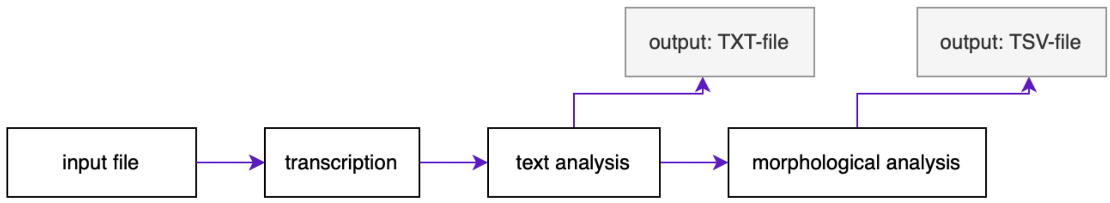

Figure 2: Flowchart of the final pipeline.

3.1.1 Subject-Verb Agreement

The first function, `subject_verb_agreement()`, checks whether the person and number of a subject and its corresponding verb match. To do so, the user-corrected transcription of the audio file is sentence-tokenised (split into sentences) using spaCy. Every sentence in the text is then word-tokenised (split into words). To find the subject of the sentence, all tokens are searched for the part-of-speech (POS) tags '`sb`', '`cd`' and '`cj`'. The latter two indicate a conjunction in subjects like *sie und ich* (English: 'she and I'). In the same step, the verb of the sentence is extracted and added to the tuple. The loop thus creates a list of all subject-verb tuples. Once the list is complete, the function runs over each subject-verb tuple and checks for various morphological attributes. See Figure 3 for an overview of the function.

I had to consider many different edge cases to check for subject-verb agreement. For example, the length of the subject (in number of tokens) could vary. A first name or pronoun is a single token, whereas a conjunction consists of multiple tokens. Considering all possibilities is necessary to yield results as precise as possible. If the subject is only one token, I check for its POS tag. If it is a pronoun, it may need to be disambiguated. In German, the female third-person singular is *sie*, which is identical to the third-person plural (English: 'she' and 'they'). Due to such ambiguities, spaCy is not able to tag perfectly. Thus, manual disambiguation is required to prevent downstream errors. The function checks for agreement in person and number between the subject and the verb. If all four morphological attributes match, the tuple is assumed to be correct. If any attribute does not match, yet further disambiguation might be required. For example, the German *ihr* can either be the second-person plural pronoun (English: '*you* seem like very happy people') or a possessive pronoun for the female third-person singular (English: '*her* friend'). For subjects longer than one token, the function first searches for conjunctions. If one is present, number and person agreement are checked for both the subject and the verb. If no conjunction is present, such as in the construction *unsere beiden Patientinnen* (English: 'our two patients'), the core subject-token is extracted (in this case, *Patientinnen*, thus 'patients'). The same checks are applied to this core subject-token. If no subject can be found at all, I assume either an error by spaCy or a grammatically incorrect sentence. Such cases will be marked as an error with the message '`ERROR BY SPACY`'.

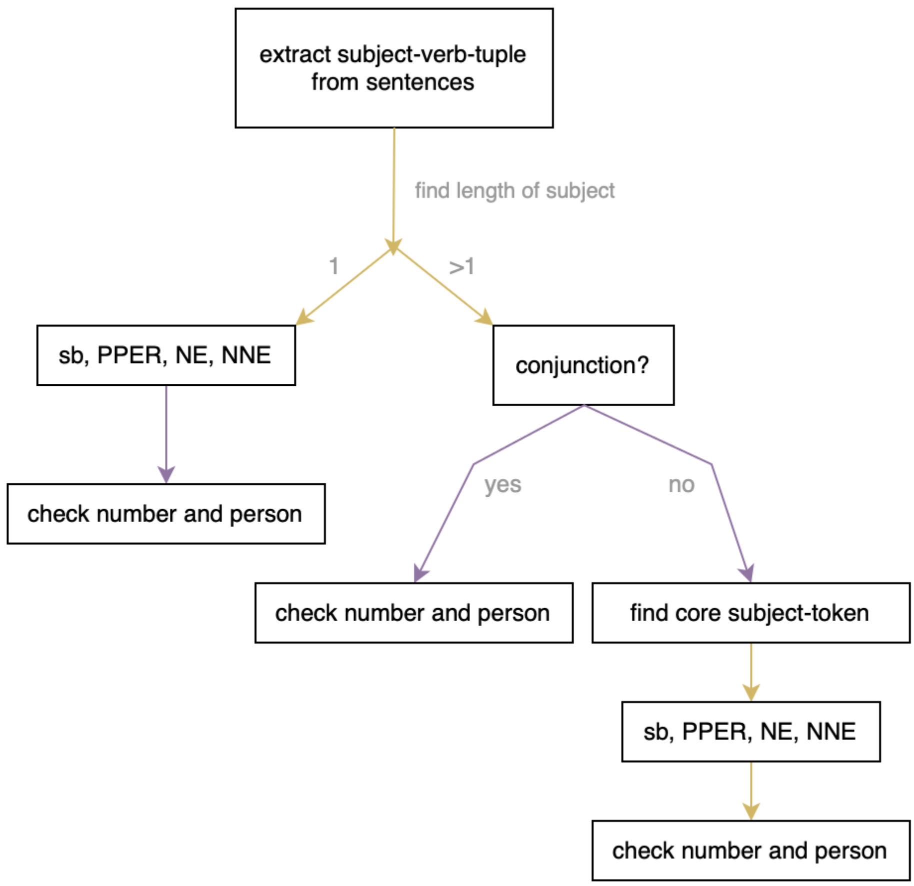

Figure 3: Flowchart of the function `subject_verb_agreement()`.

The information for the TSV file is simultaneously gathered and stored. The function returns a list of lists, which each contain one line of the TSV file. The format and data of the TSV file lines are explained in the following section, 3.1.3 Format of TSV Files.

3.1.2 Formation of Plurals

The second function, `wrong_plurals()`, checks the plural forms in the text by looking them up on a website. Originally, I wanted to use LEO.org[1]. Since they mostly offer translation services and not look-up services, however, I switched to Cactus2000.de. This site offers a large dictionary and the necessary declension information. See the screenshot in Figure 5 for an example.

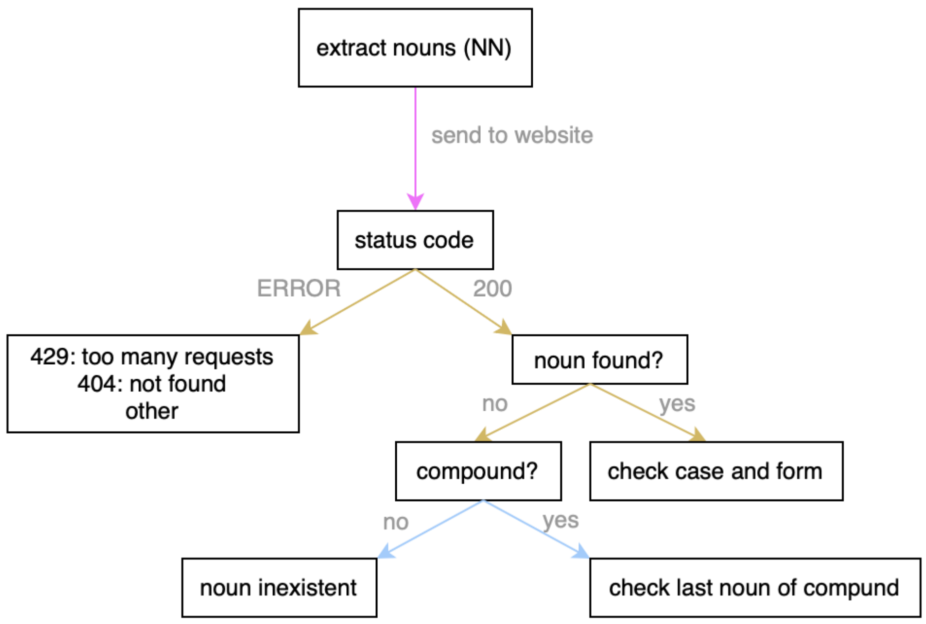

Figure 4: Flowchart of the function `wrong_plurals()`.

My function accesses this website for each noun in the text and checks its plural form. The flowchart in Figure 4 illustrates this process. One disadvantage of this method is that any future changes on the website will create problems. Using the module `webdriver`, the code accesses specific items and buttons on the website. If a name or ID changes in the script of the website, my code would have to be rewritten at least in parts. The use of an API would be more robust, but I unfortunately could not find an appropriate one for German.

Aside from the longevity issue, the function works well. After all nouns have been extracted, each of them is sent to the website. If the request is successful (`status code 200`), the token from the text is searched in the table. If it exists and the morphological cases match, the form is assumed to be correct. If it does not match, it is appended to a list which will be used in writing the TSV file. In case the noun from the original text is not found on the website, there are two options. The noun

[1]Click to be redirected to the websites mentioned.

die Sprache [f]

	Singular	Plural
Nom.	**die Sprache**	die Sprachen
Gen.	**der Sprache**	der Sprachen
Dat.	**der Sprache**	den Sprachen
Akk.	**die Sprache**	die Sprachen

Figure 5: Declension table for the noun *Sprache* (English: 'language') from Cactus2000.de, which is accessed in the pipeline.

is either a compound form which is not in the dictionary or the noun does not exist at all. The function also catches errors from the website, like '`too many requests`' (`status code 429`) or '`not found`' (`status code 404`), and informs the user.

3.1.3 Format of TSV Files

As mentioned before, the goal of the pipeline is not only widely accessible screening of children's speech, but also the generation of data which can be used to train a ML model. To provide data for increased machine-readability and processability, I decided to store the output in a tab-separated-values file (TSV). The user can choose to store the output of only one analysis (either checking for subject-verb agreement or wrong plurals) or of both.

For both types of analysis, the format of the TSV file is the same. Each function returns a list of lists of length five containing information for each sentence. This list is then put through the helper function `list_to_tsv()`, which returns the list as a string separated by tabs. Finally, the output from the helper function is written to the final TSV file line by line. The layout for one analysis per TSV file is as follows:

```
id   errored   count_err   errors   sentence
```

The `id` enumerates the sentences from the original text, starting with `0`. The second column of the TSV file is of a binary nature and can have the value `0` or `1`. It indicates whether the current sentence has any errors or not, with `0` indicating no errors and `1` indicating at least one error. The third column is the count of errors in the current sentence, and the fourth enlists them in a string. For `subject_verb_agreement()`, the subject-verb tuples are stored as follows:

```
subject1-verb1/subject2-verb2/...
```

`wrong_plurals()` is very similar:

```
noun1-case1/noun2-case2/...
```

The last column of the file contains the full sentence as a string. This can be helpful for further analysis or for reconstructing the error analysis.

If the results of both analyses are combined in the TSV file, the layout is slightly adapted, though necessarily more complicated. Each line of the TSV file looks like the following:

```
id  errored  sva_err_count  sva_errors  pl_err_count  pl_errors
sentence
```

This format of the TSV file allows important information to be stored and kept easily readable for a machine. Additionally, programming in Python with a TSV file can be done well and allows further implementations using the generated data.

3.2 Results

To run the pipeline, users only need to run the script from the command line via Python. Once all requirements have been installed, the user can type

```
$ python3 pipeline.py
```

to the command line. The pipeline continues until the morphological analysis has been done. The user only has to upload the MP3 file to be transcribed and analysed. To demonstrate the workings of the functions, I show some examples of the outputs.

3.2.1 Function subject_verb_agreement()

As mentioned, the function `subject_verb_agreement()` extracts subject-verb tuples and checks for morphological congruence. For example, for the sentence `Die Kinder gehen gerne zur Schule.`[2], the tuple `([Die, Kinder], gehen)` is extracted. From the tuple's first element, the core subject-token is searched. In this case, it would be the noun *Kinder*. Once it has been found, the analysis compares its morphology to the verb's. Since this sentence is grammatically sound, the output of the function is the list

[2]English: 'The children enjoy going to school.'

```
[[0, 0, 0, '', 'Die Kinder gehen gerne zur Schule.']]
```

The output looks very simple if there are no errors in the sentence. The next example contains a sentence with an incorrect verb form. Instead of *wollen* (English: verb 'want') in the third-person singular, the second-person plural form is written: `Der Junge wollte ihn haben und dann der Junge wollt ihn holen.`[3] The extracted subjects will be the tuples

```
[([Junge], wollte), ([Junge], wollt)]
```

stored in a list. The analysis is applied to the list and returns the following:

```
[[0, 0, 1, 'Junge-wollt/', 'Der Junge wollte ihn haben und dann
der Junge wollt ihn holen.']]
```

The error is stored as a string containing the subject and the verb. For passive constructs in sentences like `Der Hund werden vom Kind gestreichelt.`[4], the subject and verb are identified correctly and the function returns the error:

```
[[0, 1, 1, 'Der Hund-werden/',
'Der Hund werden vom Kind gestreichelt.']]
```

3.2.2 Function wrong_plurals()

The function `wrong_plurals()` returns a very similar output to the function shown above. It searches all nouns in a sentence and looks up the plural forms. For example, the sentence `Die Hunden werden von den Kindern gerufen.`[5] has the nouns *Hund* (English: 'dog') and *Kind* (English: 'child'). The former is in the wrong form and is thus returned in the output:

```
[[0, 1, 1, 'Hunden-Nom/', 'Die Hunden werden von den Kindern
gerufen.']]
```

The function works for common nouns only. Named entities or proper nouns such as person names, are not always analysed properly especially since plurals for names are not very common. For example, the sentence `Alle Lisas sollen nach vorne kommen.`[6] will not yield any errors, but if the plural is changed to an incorrect form such as 'Lisen', it will return `No such noun "Lisen" found.`

[3] English: 'The boy wanted to have it and then the boy wanted to get it.'

[4] English: 'The dog are pet by the child.'

[5] English: 'The dogs are called by the children.'

[6] English: 'All Lisas should come to the front.'

3.3 Evaluation

To see how well my extension works in a 'real-life' application, I completed an evaluation on a sample audio file. The data is a voice recording of a German-speaking, multilingual child with a DLD provided by Susanne Kempe from the University of Teacher Education in Special Needs. Since the recording used for my evaluation is not available publicly, I added a sample audio file from Brauer [2014] to the GitLab repository for testing the code.

To evaluate my code, I created my own annotation of the final transcription and compared it to the pipeline's output. I transcribed the audio file using the Prodigy interface, but skipped the rest of the pipeline. Adjusting the automatic transcription was already a first challenge, since young children's speech tends to include pauses, stutters, and hesitations and is generally not very fluent. Furthermore, the text has to be transcribed in a way that can be easily handled by spaCy. I decided to use dashes for the child correcting themself. For example[7],

`Und der Ball ist hinter den - der Hund ist hinter den Ball.`[8]

To transcribe an unfinished sentence without any corrections made by the child, I used an ellipsis. For example:

`Da hat er den... hm.`[9]

I manually split the transcription into 74 sentences. I checked for errors in both subject-verb agreement and plural formation in each sentence. The final transcription consists of 145 sentences, four of which contain errors when analysed properly. All of these are due to incorrect subject-verb agreement.

Both analyses split the text into 74 sentences and 145 possible errors. In Table 2 on page 18, I gather my annotations and the pipeline's. Cells with positive agreement or disagreement are coloured. No colour is shown for the sentences where both I and the pipeline annotated 0. Agreement is coloured blue, disagreement is coloured magenta. Cells containing more than one value represent the possible errors in a single sentence. For example, sentence (9) in the transcript:

[7]The examples provided are from the publicly available audio file after being analysed with the pipeline. The original evaluation was done on a file which is not publicly available - I can thus only provide the results and not the data itself. The transcription of the sample file can be found in Appendix A.

[8]English: 'And the ball is behind the – the dog is behind the ball.'

[9]English: 'Then he has the... hm.'

`Und dann geht der Junge runter, weil der Hund ganz klein ist.`[10]

has two conjugated verbs, which makes two possible errors for subject-verb agreement.

		Manual		
		1	0	Total
Automatic	1	3	21	24
	0	1	110	111
	Total	4	131	145

Table 1: Matrix to calculate inter-annotator agreement.

To put Table 1 in numbers, the automatic analysis and my analysis match in 113 out of 145 cases, which is an observed agreement of 78%. When calculating inter-annotator agreement (IAA), Cohen's Kappa is 0.275. On the scale introduced by lan, the score is only 'slight'.

3.4 Discussion

Creating a pipeline for automatic annotation of children's speech is a substantial task that needs to be split into sub-parts. For my thesis, I implemented an automatic morphological analysis which checks subject-verb agreement and the formation of plurals. Running the pipeline prompts the user to input an audio file which is transcribed automatically. After the user manually corrects the transcription, the lexical and morphological aspects of the final text are analysed. Finally, the user receives files with the transcriptions and the results of the analysis. These files can be taken as they are or used for further analysis.

To examine how well the code performs, I evaluated 74 sentences with 145 possible errors. Between manual and automatic annotation there is 78% observed agreement. Calculating the inter-annotator agreement yielded a Cohen's Kappa of 0.275, which is 'slight' according to the scale by lan. This low score can be explained - the evaluation was done on a small sample, which is not representative of the overall quality of implementation. The evaluation has been done manually due to limited resources, which also limits the size of the testing data. Regardless, comparing

[10]English: 'And then the boy goes down because the dog is very small.

manual with automatic annotation can give an insight into how well the system works.

The low Cohen's Kappa score of 0.275 is presumably due to the small number of samples and some incorrect tagging by spaCy. A broader evaluation with more data may have yielded better results. One recurring noun in this particular sample resulted in the same error multiple times. Again, since the sample size is so small, such errors account for much more of the overall evaluation.

Implementation could be improved in various aspects. Firstly, the current version is brittle as it accesses websites using `webdriver` and `selenium`. While very useful, these modules are limited in their applicability. As soon as the name or ID of a website's element changes, the element becomes inaccessible and the code must be debugged manually. A better solution would be to use an API (Application Programming Interface). Unfortunately, neither IMS-Speech nor the website used to look up plural forms currently offer such an interface. An API to look up German nouns would be a significant improvement. The website itself (Cactus2000.de) is also limited in its suitability for this application. Finally, spaCy is a very reliable tool for tagging and parsing grammatically correct language, but struggles much more with non-grammatical language. Children's disordered speech in particular often lacks syntax and contains many repetitions due to hesitations and pauses. A different tagger better suited to this type of data may improve performance.

Despite these issues, the pipeline offers some substantial benefits. It does not require the user to have any programming skills, which makes it more accessible to therapists. Now that Prodigy supports spaCy, there is no longer any need to change virtual environments as in the previous version of the pipeline.

Index	SVA_P	SVA_S	PL_P	PL_S	Index	SVA_P	SVA_S	PL_P	PL_S
0	1	0			37			0	0
1	1	1			38	0	0	0	0
2	0	0	0	0	39			1	0
3	0, 0	0, 0			40	0, 0	0, 0	1, 0	0, 0
4	0	0			41				
5	0	0			42	0	0	1	0
6	0	0			43	0	0		
7	0	0			44			0, 0	0, 0
8	0	0	0	0	45	1, 0	0, 0	0	0
9	0	0			46	0	0	0	0
10	0, 0, 0, 0	0, 0, 0, 0	0, 0	0, 0	47	0	0	1	0
11	0, 0	0, 0	1, 0	0, 0	48	0	0		
12	0	0	1	0	49	0, 0	1, 0	1, 0	0, 0
13	1, 0	0, 0	1	0	50	0, 0	0, 0	1, 0, 0	0, 0, 0
14	0	0			51				
15	0	0			52	0	0	0, 0	0, 0
16	0	0	0	0	53	0	0		
17	1	1	0, 0	0, 0	54	0	0	0	0
18	0	0	0	0	55	0, 0	0, 0		
19	0, 0, 0	0, 0, 0			56	0	0		
20	1	0	0, 0, 0, 1	0, 0, 0, 0	57	0, 0, 1	0, 0, 1	1, 0, 0	0, 0, 0
21	0	0	0	0	58	0	0	0	0
22					59	0, 0	0, 0	0	0
23	0	0	0	0	60	0	0	0	0
24					61	0	0	0	0
25	0	0			62	0	0	0	0
26	0, 0	0, 0	0	0	63	0	0	1	0
27	0	0	0, 0	0, 0	64	0	0		
28	0	0	0, 0	0, 0	65				
29			0	0	66	0	0		
30	0, 0	0, 0			67				
31	0	0	1, 0, 0	0, 0, 0	68				
32	0	0			69	0	0		
33	0	0	0	0	70	0, 0, 0	0, 0, 0	0, 0	0, 0
34	0, 0	0, 0	1	0	71	0	0	0	0
35	0	0			72	1	0		
36	0	0	1	0	73			1	0

Table 2: Errors annotated by the pipeline (P) and by me (S). Blue: positive agreement, magenta: disagreement.

4 Conclusion and Future Work

Developmental Language Disorder (previously referred to as 'Specific Language Impairment') affects up to an estimated 15% of German-speaking children. Early diagnosis and intervention can significantly improve a child's language skills. If the Developmental Language Disorder remains undiagnosed, the child may never fully learn their mother tongue and face additional challenges in their social life and school performance.

Developmental Language Disorder can be diagnosed based on many different aspects of a person's speech. These can be split into four linguistic levels: lexicon and semantics, syntax and morphology, phonology, and pragmatics. Monitoring the different levels carefully over an extended period of time can yield a good diagnosis. Since this is very costly in both time and money, however, an automated approach would be desirable. Research on this topic is very active in the field of machine learning. A model that can predict a child's language development based on a speech recording would enormously increase the number of children who can be screened. The pipeline implemented creates the necessary data to train such a model data and can aid therapists in pre-diagnosing many children in a short amount of time.

The extension of the pipeline (code can be found on GitLab including a sample audio file) focuses on morphological analysis: more specifically, subject-verb agreement and the formation of plurals. Errors in both aspects are strong signs for a Developmental Language Disorder. Initial results from the pipeline look promising, although the current version stands to be improved in some ways. Cohen's Kappa between manual and automatic annotation on a small sample is 0.275.

The pipeline offers many opportunities for improvement and further work. Overall, the morphological analysis itself can be refined. The pipeline still struggles to annotate very well, since it relies on spaCy's tagging. SpaCy, in turn, struggles with disordered speech which can yield unreliable results. As mentioned in Chapter 3, one option would be to use a different tagger or to create a task-specific one to use on disordered speech.

Further, the morphological analysis can be extended. There are many different aspects of speech which could be monitored. Enriching the analysis will return more accurate results. Analysing the speech on different levels may also increase the accuracy and specifity of the pipeline. For example, a syntactic or pragmatic analysis could be implemented.

For now, the pipeline works for German and is highly language-specific. Further work could include extending the pipeline to other dialects of German, since children are more likely to speak their dialect than Standard German. SpaCy's German model is trained to perform best on grammatical Standard German, which many German-learning children do not speak. The pipeline could also be extended to other languages. This flexibility would positively affect the usability of the pipeline.

To enhance the overall user experience, a graphical user interface could be created. A website or application would be easier to work with than a command line interface, particularly for clinicians and other professionals without coding skills who research and work with children with DLD. This would improve accessibility and user-friendliness. Therapists will use the interface more and be able to screen more children, with the additional benefit of simultaneously creating more training data for a machine learning model. Such a pipeline not only offers the opportunity to screen many children and diagnose Developmental Language Disorder as early on as possible, but it also further increases our knowledge of DLD in general. Gathering numerous cases of DLD will impact general understanding of it in a positive manner. Additionally, such a model creates data, which could be used for training other models, getting us closer to a fully automated, precise, and efficient diagnosis.

References

D. Bishop, M. Snowling, P. Thompson, T. Greenhalgh, M. Westerveld, C.-A. Murphy, B. McNeill, and A. Whitehouse. Phase 2 of CATALISE: A multinational and multidisciplinary Delphi consensus study of problems with language development: Terminology. *Journal of Child Psychology and Psychiatry*, 58, 03 2017. doi: 10.1111/jcpp.12721.

T. Brauer. *Logopädie: Was ist das?* Schulz-Kirchner Verlag, Idstein, 5th edition, 2014. ISBN 9783824803644.

J. Clegg, C. Hollis, L. Mawhood, and M. Rutter. Developmental language disorders - A follow-up in later adult life. Cognitive, language and psychosocial outcomes. *Journal of child psychology and psychiatry, and allied disciplines*, 46: 128–49, 03 2005. doi: 10.1111/j.1469-7610.2004.00342.x.

S. Conrad. Toward automatic diagnosis of medical conditions involving speech impairment, 2021.

U. de Langen-Müller, C. Kauschke, and K. Neumann. *Diagnostik von (umschriebenen) Sprachentwicklungsstörungen*. Peter Lang Verlag, Berlin, Deutschland, 2012. URL `https://www.peterlang.com/document/1046009`.

P. Denisov and N. T. Vu. IMS-Speech: A Speech to Text Tool, 2019. URL `https://arxiv.org/abs/1908.04743`.

A. Fernald, V. Marchman, and A. Weisleder. SES Differences in Language Processing Skill and Vocabulary are Evident at 18 Months. *Developmental science*, 16:234–48, 03 2013. doi: 10.1111/desc.12019.

P. Grill and J. Tuckova. Speech Databases of Typical Children and Children with SLI. *PLOS ONE*, 11:e0150365, 03 2016. doi: 10.1371/journal.pone.0150365.

H. Grimm, M. Aktas, T. Jungmann, S. Peglow, D. Stahn, and E. Wolter. Sprachscreening im Vorschulalter: Wie viele Kinder brauchen tatsächlich eine Sprachförderung? *Frühförderung interdisziplinär*, 23(3):108–117, 2004.

C. Kauschke. doi: 10.1515/9783110283891.

J. F. Miller, K. Andriacchi, and A. Nockerts. *Assessing Language Production Using SALT Software: A clinician's guide to language sample analysis (3rd Edition)*. SALT Software, LLC., 2019.

T. Solorio. Survey on Emerging Research on the Use of Natural Language Processing in Clinical Language Assessment of Children. *Lang. Linguistics Compass*, 7:633–646, 2013.

W. Suchodoletz. Umschriebene Sprachentwicklungsstörungen. *Monatsschrift Kinderheilkunde*, 151:31–37, 01 2003. doi: 10.1007/s00112-002-0644-3.

Curriculum Vitae

Personal

Sofia Carpentieri
Loorenstrasse 45
8053 Zurich
sofia.carpentieri@uzh.ch

Education

2007 - 2013	Primary School at Schulhaus Fluntern, Zurich (CH)
2013 - 2019	Secondary School at Literargymnasium Rämibühl, Zurich (CH)
since 2019	Bachelor Computational Linguistics and Speech Technology at the University of Zürich (CH)

Teaching at the University

Spring 2021	Tutorial *Introduction to Computational Linguistics 2*
Fall 2021	Tutorial *Programming for Linguists*
Spring 2022	Tutorial *Programming Techniques in Computational Linguistics 2*

A Final Transcription

This section contains the transcription of the sample audio file (Brauer [2014]) which can be found on GitLab. The examples provided in the evaluation are taken from this transcript, as the original file used is not publicly available.

```
Das Mädchen wirft auf dem Heft - auf den Ball zu.  Das Mädchen
wirft auf den Ball zu auf den Jungen.  Dann hat der Junge nicht
gefangen.  Dann ist der, das ist der Ball sehr geflogen auf den
hinter den Hau- auf den hinter den Zaun.  Und dann hat der -
da hat der Junge...  Dann ist der Ball hinter den Zaun.  Der
Junge wollte ihn haben und dann der Junge wollte ihn holen.  Da
hat er den...  hm.  Mädchen festhalte ihn.  Und dann geht der
Junge runter, weil der Hund ganz klein ist.  Und das Mädchen
guckt einfach den Hund an.  Und der Ball ist hinter den - der
Hund ist hinter den Ball.  Dann gibt er den alter Opa.  Dann
gibt er den alten Opa.  Kommt der alte Opa raus, dann gibt er
den Junge den Ball und das Mädchen freut sich und der Junge auch.
```