Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2706–2711
Marseille, 11–16 May 2020

Automatic Creation of Text Corpora for Low-Resource Languages from the Internet: The Case of Swiss German

Lucy Linder[1], Michael Jungo[1], Jean Hennebert[1], Claudiu Musat[3], Andreas Fischer[1,2]
[1] iCoSys Institute, University of Applied Sciences and Arts Western Switzerland
[2] DIVA Group, University of Fribourg, Switzerland
[3] Data, Analytics and AI Lab, Swisscom AG, Switzerland
{lucy.linder, michael.jungo, jean.hennebert, andreas.fischer}@hefr.ch, claudiu.musat@swisscom.com

Abstract

This paper presents SwissCrawl, the largest Swiss German text corpus to date. Composed of more than half a million sentences, it was generated using a customized web scraping tool that could be applied to other low-resource languages as well. The approach demonstrates how freely available web pages can be used to construct comprehensive text corpora, which are of fundamental importance for natural language processing. In an experimental evaluation, we show that using the new corpus leads to significant improvements for the task of language modeling. To capture new content, our approach will run continuously to keep increasing the corpus over time.
Keywords: Corpus, Language Identification, Less-Resources/Endangered Languages, Language Modeling, Tools

1. Introduction

Swiss German ("Schwyzerdütsch" or "Schwiizertüütsch", abbreviated "GSW") is the name of a large continuum of dialects attached to the Germanic language tree spoken by more than 60% of the Swiss population (Coray and Bartels, 2017). Used every day from colloquial conversations to business meetings, Swiss German in its written form has become more and more popular in recent years with the rise of blogs, messaging applications and social media. However, the variability of the written form is rather large as orthography is more based on local pronunciations and emerging conventions than on a unique grammar.

Even though Swiss German is widely spread in Switzerland, there are still few natural language processing (NLP) corpora, studies or tools available (Hollenstein and Aepli, 2014). This lack of resources may be explained by the small pool of speakers (less than one percent of the world population), but also the many intrinsic difficulties of Swiss German, including the lack of official writing rules, the high variability across different dialects, and the informal context in which texts are commonly written. Furthermore, there is no official top-level domain (TLD) for Swiss German on the Internet, which renders the automatic collection of Swiss German texts more difficult.

To foster the development of NLP tools for Swiss German, we gathered the largest corpus of written Swiss German to date by crawling the web using a customized tool. We highlight the difficulties for finding Swiss German on the web and demonstrate in an experimental evaluation how our text corpus can be used to significantly improve an important NLP task : language modeling.

2. Related Work

Few GSW corpora already exists. Although they are very valuable for research on specific aspects of the Swiss German language, they are either highly specialized (Stark et al., 2009) (Samardžić et al., 2016) (Grubenmann et al., 2018), rather small (Hollenstein and Aepli, 2014) (7,305 sentences), or do not offer full sentences (Scannell, 2007).

To our knowledge, the only comprehensive written Swiss German corpus to date comes from the Leipzig corpora collection initiative (Goldhahn et al., 2012) offering corpora for more than 136 languages. The Swiss German data has two sources: the Alemannic Wikipedia and web crawls on the `.ch` domain in 2016 and 2017, leading to a total of 175,399 unique sentences. While the Leipzig Web corpus for Swiss German is of considerable size, we believe this number does not reflect the actual amount of GSW available on the Internet. Furthermore, the enforced sentence structures do not represent the way Swiss German speakers write online.

In this paper, we thus aim at augmenting the Leipzig Web corpus by looking further than the `.ch` domain and by using a suite of tools specifically designed for retrieving Swiss German.

The idea of using the web as a vast source of linguistic data has been around for decades (Kilgarriff and Grefenstette, 2003) and many authors have already addressed its importance for low-resources languages (Ghani et al., 2001). A common technique is to send queries made of mid-frequency n-grams to a search engine to gather bootstrap URLs, which initiate a *crawl* using a breadth-first strategy in order to gather meaningful information, such as documents or words (Sharoff, 2006), (Scannell, 2007).

Existing tools and studies, however, have requirements that are inadequate for the case of Swiss German. For example, GSW is not a language known to search engines (Sharoff, 2006), does not have specific TLDs (Schäfer and Bildhauer, 2012), and lacks good language identification models. Also, GSW documents are too rare to use bootstrapping techniques (Ghani et al., 2001). Finally, as GSW is scarce and mostly found in comments sections or as part of multilingual web pages (e.g. High German), we cannot afford to "privilege precision over recall" (Baroni and Kilgarriff, 2006) by focusing on the main content of a page.

As a consequence, our method is based on known techniques that are adapted to deal with those peculiarities. Furthermore, it was designed for having a human in the loop. Its iterative nature makes it possible to refine each step of the tool chain as our knowledge of GSW improves.

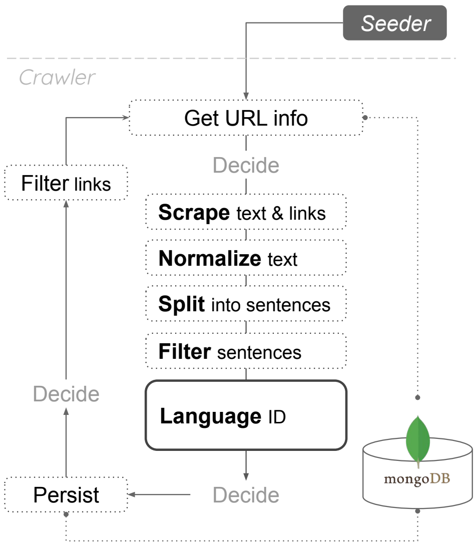

Figure 1: Overview of one iteration of the proposed system.

3. Proposed System

The two main components of our proposed system are shown in Figure 1: a *seeder* that gathers potentially interesting URLs using a Search Engine and a *crawler* that extracts GSW from web pages, linked together by a MongoDB database. The system is implemented in Python 3, with the full code available on GitHub[1]. Due to the exploratory nature of the task, the tool chain is executed in an iterative manner, allowing us to control and potentially improve the process punctually.

3.1. The Seeder

Query generation has already been extensively studied (Baroni and Ueyama, 2006), (Sharoff, 2006). In the case of Swiss German, we tested three different approaches: (a) most frequent trigrams, (b) selection of 2 to 7 random words weighted by their frequency distribution and (c) human-generated queries.
When comparing the corpora generated by 100 seeds of each type, we did not observe significant differences in terms of quantity or quality for the three seeding strategies. On a positive side, 50% of the sentences were different from one seed strategy to the other, suggesting for an approach where strategies are mixed. However, we also observed that (a) tends to yield more similar queries over time and (c) is too time-consuming for practical use.
Considering these observations, we privileged the following approach:

1. Start with a list of sentences, either from a bootstrap dataset or from sentences from previous crawls using one single sentence per unique URL;

2. Compute the frequency over the vocabulary, normalizing words to lower case and discarding those having non-alphabetic characters;

[1] `https://github.com/derlin/swisstext-lrec`

3. Filter out words appearing only once or present in German or English vocabularies[2];

4. Generate query seeds by sampling 3 words with a probability following their frequency distribution;

5. Exclude seeds with more than two single-letter words or having a GSW probability below 95% (see the language identification (LID) component, Section 3.3.).

Initial sentences come from the Leipzig web corpus 2017, filtered by means of the LID described in Section 3.3.
Each seed is submitted to startpage.com, a Google Search proxy augmented with privacy features. To ensure GSW is not auto-corrected to High German, each word is first surrounded by double quotes. The first 20 new URLs, i.e. URLs that were never seen before, are saved for further crawling.

3.2. The Crawler

The crawler starts with a list of URLs and metadata taken either from a file or from the MongoDB instance, and are added to a task queue with a depth of 0. As illustrated in Figure 1, each task consists of a series of steps that will download the page content, extract well-formed GSW sentences and add links found on the page to the task queue. At different stages of this pipeline, a *decider* can intervene in order to stop the processing early. A crawl may also be limited to a given depth, usually set to 3.

Scrape The raw HTML content is fetched and converted to UTF-8 using a mixture of `requests` and `BeautifulSoup`. Boilerplate removal such as navigation and tables uses `jusText` (Pomikálek, 2011), but ignores stop words filtering as such a list is not available for GSW. The output is a UTF-8 text.

Normalize This stage tries to fix remaining encoding issues using `ftfy` (Speer, 2019) and to remove unicode emojis. Another important task is to normalize the unicode code points used for accents, spaces, dashes, quotes etc., and strip any invisible characters. To further improve the usability of the corpus and to simplify tokenization, we also try to enforce one single convention for spaces around quotes and colons, e.g. colons *after* closing quote, no space inside quotes.

Split To split text into sentences, we implemented Moses' `split-sentences.perl`[3] in Python and changed it in three main ways: existing newlines are preserved, colons and semi-colons are considered segmentation hints and sentences are not required to start with an uppercase. The latter is especially important as GSW is mostly found in comments where people tend to write fast and without proper casing/punctuation. The list of non-breaking prefixes used is a concatenation of the English and German prefixes found in Moses with few additions.

[2] Free German Dictionary, `https://sourceforge.net/projects/germandict/` 1.9M words and US word list from GNU Aspell `http://aspell.net/`, 40K words.
[3] `https://github.com/moses-smt/mosesdecoder`.

	Seeding				*Crawling*			
Iter.	Seeds	Found	Good	%Good	Sentences	Domains	URLs	Runtime
0	100	837	577	68.94	89350	529	4810	16h15
1	100	1062	662	62.34	48483	552	4382	16h52
2	100	732	423	57.79	20662	423	2193	40h52

Table 1: Number of URLs found vs actually pertinent during seeding and number of unique new sentences, domains, and URLs discovered during crawl starting from a blank database and launching the system three times.

Filter Non- or bad- sentences are identified based on a list of 20+ rules that normal sentences should obey. Most rules are specified in the form of regular expression patterns and boundaries of acceptable occurrences, few compare the ratio of occurrence between two patterns. Examples of such rules in natural language are: "no more than one hashtag", "no word with more than 30 characters", "the ratio capitalized/lowercase words is below 1.5".

Language ID Using the language identification model described in Section 3.3., sentences with a GSW probability of less than 92% are discarded. This threshold is low on purpose in order to favor recall over precision.

Link filter This component is used to exclude or transform outgoing links found in a page based on duplicates, URL composition, but also specific rules for big social media sites or known blogs. Examples are the exclusion of unrelated national TLDs (`.af`, `.nl`, `...`) and known media extensions (`.pdf`, `.jpeg`, etc.), the stripping of session IDs in URL parameters, and the homogenization of subdomains for sites such as Twitter. Note that filtering is based only on the URL and therefore does not handle redirects or URLs pointing to the same page. This leads to extra work during the crawling, but keeps the whole system simple.

Decide A decider has three main decisions to take. First, based on the metadata associated with an URL, should it be visited? In practice, we visit only new URLs, but the tool is designed in a way such that recrawls would be triggered depending on how much new content is found over time. The second decision arises at the end of the processing, where the page can be either saved or blacklisted. To favor recall, we currently keep any URL with at least one GSW sentence. Finally, the decider can choose to visit the outgoing links or not. After some trials, we found that following links from pages with more than two new GSW sentences is a reasonable choice, as pages with less sentences are often quotes or false positives.

Duplicates During the crawl, the uniqueness of sentences and URLs considers only exact matches. However, when exporting the results, near-duplicate sentences are removed by first stripping any non-letter (including spaces) and making a lowercase comparison. We tried other near-duplicate approaches, but found that they also discarded meaningful writing variations.

3.3. Language Identification

Language identification (LID) is a central component of the pipeline, as it has a strong influence on the final result. In addition, readily available tools are not performing at a satisfying level. For these reasons we created a tailor-made LID system for this situation.

Label	Languages
AFR	Afrikaans
DEU	German
ENG	English
GSW	Swiss German
GSW_LIKE	Bavarian, Kölsch, Limburgan, Low German, Northern Frisian, Palatine German
LTZ	Luxembourgian
NLD	Dutch
OTHER	Catalan, Croatian, Danish, Esperanto, Estonian, Finnish, French, Irish, Galician, Icelandic, Italian, Javanese, Konkani, Papiamento, Portuguese, Romanian, Slovenian, Spanish, Swahili, Swedish

Table 2: Composition of the eight classes used for language identification.

LID has been extensively studied over the past decades (Jauhiainen et al., 2019) and has achieved impressive results on long monolingual documents in major languages such as English. However, the task becomes more challenging when the pool of training data is small and of high variability, and when the unit of identification is only a sentence.

Free pretrained LIDs supporting GSW such as FastText (Joulin et al., 2016) are trained on the Alemannic Wikipedia, which encompasses not only GSW, but also German dialects such as Badisch, Elsässisch, Schwäbisch and Vorarlbergisch. This makes the precision of the model insufficient for our purposes.

The dataset used to build our Swiss German LID is based on the Leipzig text corpora (Goldhahn et al., 2012), mostly focusing on the texts gathered from the Internet. In preliminary experiments, we have chosen eight language classes shown in Table 2, which give precedence to languages closely related to Swiss German in their structure. In this Table, `GSW_LIKE` refers to a combination of dialects that are similar to Swiss German but for which we did not have sufficient resources to model classes on their own.

A total of 535,000 sentences are considered for LID with an equal distribution over the eight classes. The 66,684 GSW sentences originate from the Leipzig web corpus 2017 and have been refined during preliminary experiments to exclude obvious non-GSW contents. We use 75% of the data

for training, 10% for optimizing system parameters, and 15% for testing the final performance.
Using a pretrained German BERT model (Devlin et al., 2018)[4] and fine-tuning it on our corpus, we obtain a high LID accuracy of 99.58%. GSW is most confused with German (0.04%) and `GSW_LIKE` (0.04%). We have also validated the LID system on SMS sentences (Stark et al., 2009), where it proves robust for sentences as short as five words.

4. State of the Swiss German Web

Table 1 shows the results of running the system three times using 100 different seeds on a virtual machine with 5 CPU cores and no GPUs. As expected, the first iteration yields the most new sentences. Unfortunately, the number of newly discovered hosts and sentences decreases exponentially as the system runs, dropping to 20K sentences on the third iteration. This result emphasizes the fact that the amount of GSW on the web is very limited.
The third iteration took also significantly longer, which highlights the difficulties of crawling the web. In this iteration, some URLs had as much as 12 thousand outgoing links that we had to visit before discarding. Another problem arises on web sites where query parameters are used in URLs to encode cookie information and on which duplicate hypotheses cannot be solved unless visiting the links.
On each new search engine query, we go further down the list of results as the top ones may already be known. As such, the percentage of pertinent URLs retrieved (% good, see decider description in Section 3.2.) slowly decreases at each iteration. It is however still above 55% of the retrieved URLs on the third run, indicating that further runs may still be beneficial.

5. The SwissCrawl Text Corpus

Using the proposed system, we were able to gather more than half a million unique GSW sentences from around the web. The crawling took place between September and November 2019. The corpus is available for download[5] in the form of a CSV file with four columns: `text`, `url`, `crawl_proba`, `date`, with `crawl_proba` being the GSW probability returned by the LID system (see Section 3.3.).

5.1. Contents

The corpus is composed of 562,524 sentences from 62K URLs among 3,472 domains. The top ten domains (see Table 3) are forums and social media sites. They account for 46% of the whole corpus.
In general, we consider a GSW probability of $\geq$ 99%, to be indeed Swiss German with high confidence. This represents more than 89% of the corpus (500K) (see Figure 2). The sentence length varies between 25 and 998 characters with a mean of 92 ± 55 and a median of 77 (see Figure 3), while the number of words lies between 4 and 222, with a mean of 16 ± 10 and a median of 14. This highlights a common pattern in Swiss German writings: used mostly in informal contexts, sentences tend to be short and to include many symbols, such as emojis or repetitive punctuation.
Very long sentences are usually lyrics that lack proper punctuation and thus could not be segmented properly. We however decided to keep them in the final corpus, as they could be useful in specific tasks and are easy to filter out otherwise.
Besides the normalization described in 3.2., no cleaning nor post-processing is applied to the sentences. This is a deliberate choice to avoid losing any information that could be pertinent for a given task or for further selection. As a result, the mean letter density is 80% and only 61% of sentences both start with an uppercase letter and end with a common punctuation mark (`.!?`).
Finally, although we performed no human validation *per se*, we actively monitored the crawling process to spot problematic domains early. This allowed to blacklist some domains entirely, for example those serving embedded PDFs (impossible to parse properly) or written in very close German dialects.

Domain	#URLs	#Sentences	%
www.fcbforum.ch	6,169	54,954	9.77
www.wikiwand.com	2,404	36,432	6.48
www.heiraten.ch	1,238	36,129	6.42
forum.zscfans.ch	3,844	31,125	5.53
www.celica-t23.ch	3,007	26,446	4.70
www.fczforum.ch	7,275	23,967	4.26
www.facebook.com	3,280	17,601	3.13
dict.leo.org	255	16,109	2.86
twitter.com	2,720	9,061	1.61
swizzlink.ch	1,000	7,465	1.33
other	31,329	303,235	53.91

Table 3: The top ten domains found in the corpus.

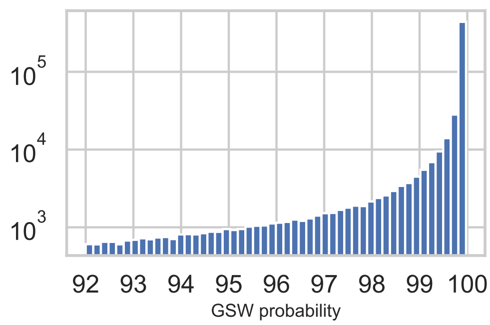

Figure 2: Distribution of `crawl_proba` (logarithmic axis).

5.2. Discussion

Table 5 shows some hand-picked examples. As most of our sources are social medias and forums, the writing style is often colloquial, interspersed with emojis and slang. This perfectly reflects the use of GSW in real life, where speakers switch to High German in formal conversations.

[4]`bert-base-german-cased` model from `https://github.com/huggingface/transformers` (Wolf et al., 2019).
[5]`https://icosys.ch/swisscrawl`

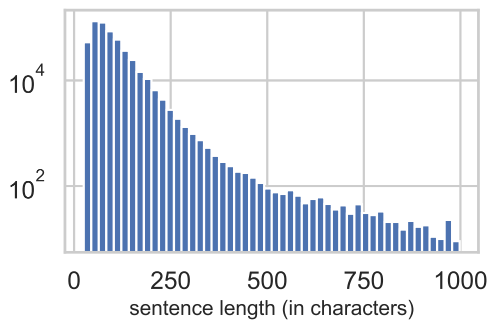

Figure 3: Distribution of `text` length (logarithmic axis).

Proba.	#Sentences	%
99+	503,526	89.51
98+	19,769	3.51
97+	11,064	1.97
96+	7,940	1.41
95+	6,367	1.13
94+	5,344	0.95
93+	4,541	0.81
92+	3,973	0.71

Table 4: Number of sentences for each 1% probability bin.

In general, the quality of sentences is good, with few false positives mostly in High German or German dialects, rarer still in Dutch or Luxembourgian. The presence of specific structures in the sentences are often the cause of such mistakes, as they yield strong GSW cues. For example:

- High German with spelling mistakes or broken words;
- GSW named entities ("Ueli Aeschbacher", "Züri");
- The presence of many umlauts and/or short words;
- The repetition of letters, also used to convey emotions.

The quality of the corpus highly depends on the text extraction step, which itself depends on the HTML structure of the pages. As there are no enforced standards and each website has its own needs, it is impossible to handle all edge cases. For example, some sites use hidden `<span>` elements to hold information, which become part of the extracted sentences. This is true for watson.ch and was dealt with using a specific rule, but there are still instances we did not detect.
Splitting text into sentences is not a trivial task. Typical segmentation mistakes come from the use of ASCII emojis as punctuation marks (see text sample 3 in Table 5), which are very common in forums. They are hard to detect due to the variability of each individual style.
We defined duplicates as having the exact same letters. As such, some sentences may differ by one umlaut and some may be the truncation of others (e.g. excerpts with ellipsis).
Finally, the corpus also contains poems and lyrics. Sometimes repetitive and especially hard to segment, they are still an important source of Swiss German online. In any case, they may be filtered out using cues in the sentence length and the URLs.

6. Swiss German Language Modeling

To demonstrate the effectiveness of the SwissCrawl corpus, we conducted a series of experiments for the NLP task of language modeling. The whole code is publicly available on GitHub[6].
Using the GPT-2 (Radford et al., 2019) model in its base configuration (12 layers, 786 hidden states, 12 heads, 117M parameters), we trained three models using different training data:

1. *Leipzig* unique sentences from the Leipzig GSW web;
2. *SwissCrawl* sentences with a GSW probability $\geq 99\%$ (see Section 5.1.);
3. *Both* the union of 1) and 2).

For each model, the vocabulary is generated using Byte Pair Encoding (BPE) (Sennrich et al., 2015) applied on the training set. BPE is a tokenization based on the most frequently occurring subword units, with the advantage of overcoming the out-of-vocabulary problem while still capturing meaningful information. The independent test sets are composed of 20K samples from each source.
Table 6 shows the perplexity of the models on each of the test sets. As expected, each model performs better on the test set they have been trained on. When applied to a different test set, both see an increase in perplexity. However, the Leipzig model seems to have more trouble generalizing: its perplexity nearly doubles on the SwissCrawl test set and raises by twenty on the combined test set.
The best results are achieved by combining both corpora: while the perplexity on our corpus only marginally improves (from 49.5 to 45.9), the perplexity on the Leipzig corpus improves significantly (from 47.6 to 30.5, a 36% relative improvement).

7. Conclusion

In this paper, we presented the tools developed to gather the most comprehensive collection of written Swiss German to our knowledge. It represents Swiss German in the way it is actually used in informal contexts, both with respect to the form (punctuation, capitalization, . . .) and the content (slang, elliptic sentences, . . .). We have demonstrated how this new resource can significantly improve Swiss German language modeling. We expect that other NLP tasks, such as LID and eventually machine translation, will also be able to profit from this new resource in the future.
Our experiments support the reasoning that Swiss German is still scarce and very hard to find online. Still, the Internet is in constant evolution and we aim to keep increasing the corpus size by rerunning the tool chain at regular intervals. Another line of future development is the customization of the tools for big social media platforms such as Facebook and Twitter, where most of the content is only accessible through specific APIs.

[6] `https://github.com/jungomi/swiss-language-model`

	Text	Domain	Proba.
1	E chlini Hommage a d Griächä, ihri kreativi Schprach und ihri relativ schrägä aber umso luschtigärä Brüch.	gurk.ch	99.96
2	aso i würd nech no bis ändi nöchscht wuche chrank schribe.	twitter.com	99.96
3	heheheh aber nunuu das pic isch geil...:-) *hützobeeeeeeee* :-D Sa 2.9.06, 10:	www.festzeit.ch	99.96
4	Super Mario Odyssey #14 - Rat wer zrugg isch...	www.youtube.com	99.40
5	14. Um(ge)kehrt ist au(ch) g'fahren. - Auerbach, Dorfgesch., III, 250;	www.zeno.org	98.91
6	"Jungfrau Zeitung - Töffli-Revival über drei Pässe", "rh":	www.google.ch	96.18

Table 5: Sample texts; 1-2 are of good quality, 3-4 contain many special characters, 5-6 are false positives (High German).

Training		*Test Sets*		
Dataset	Size	Leipzig	*SwissCrawl*	Both
Leipzig	180,000	47.6	92.6	67.6
SwissCrawl	483,526	63.9	49.5	56.2
Both	663,526	**30.5**	**45.9**	**38.0**

Table 6: Perplexity of language models trained on Leipzig, SwissCrawl (our corpus) and both.

8. References

Baroni, M. and Kilgarriff, A. (2006). Large linguistically-processed Web corpora for multiple languages. In *EACL 2006 - 11th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, pages 87–90.

Baroni, M. and Ueyama, M. (2006). Building general- and special-purpose corpora by web crawling. In *13th NIJL international symposium, language corpora: Their compilation and application*, pages 31–40.

Coray, R. and Bartels, L. (2017). Suisse allemand et allemand standard en suisse: Analyse des données de l'enquête sur la langue, la religion et la culture 2014. Technical report, Federal Statistical Office.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ghani, R., Jones, R., and Mladenic, D. (2001). Mining the web to create minority language corpora. In *CIKM*.

Goldhahn, D., Eckart, T., and Quasthoff, U. (2012). Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *LREC*, volume 29, pages 31–43.

Grubenmann, R., Tuggener, D., von Däniken, P., Deriu, J., and Cieliebak, M. (2018). Towards a corpus of swiss german annotated with sentiment. In *Proceedings of the 11th Language Resources and Evaluation Conference (LREC)*. (to appear).

Hollenstein, N. and Aepli, N. (2014). Compilation of a swiss german dialect corpus and its application to pos tagging. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 85–94.

Jauhiainen, T. S., Lui, M., Zampieri, M., Baldwin, T., and Lindén, K. (2019). Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Kilgarriff, A. and Grefenstette, G. (2003). Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29:333–347.

Pomikálek, J. (2011). *Removing Boilerplate and Duplicate Content from Web Corpora*. Ph.d. in informatics, Masaryk University, NLP Centre.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Samardžić, T., Scherrer, Y., and Glaser, E. (2016). ArchiMob - a corpus of spoken swiss German. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4061–4066, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Scannell, K. P. (2007). The crúbadán project: Corpus building for under-resourced languages. In *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop*, volume 4, pages 5–15.

Schäfer, R. and Bildhauer, F. (2012). Building large corpora from the web using a new efficient tool chain. In *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*, pages 486–493.

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural Machine Translation of Rare Words with Subword Units. *arXiv e-prints*, page arXiv:1508.07909, Aug.

Sharoff, S. (2006). Creating Specialized and General Corpora Using Automated Search Engine Queries. In *WaCky*, pages 63–98.

Speer, R. (2019). ftfy. Zenodo. Version 5.5.

Stark, E., Ueberwasser, S., and Ruef, B. (2009). Swiss sms corpus.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv, abs/1910.03771*.