Leveraging Data-Driven Methods in Word-Level Language Identification for a Multilingual Alpine Heritage Corpus

Ada Wan
University of Zurich
ada.wan@uzh.ch

Abstract

This paper presents a data-driven, simple cluster-and-label approach using optimized count-based methods for word-level language identification for a large domain-specific multilingual diachronic corpus of periodicals published at least yearly between 1864 and 2014 in Switzerland. Our system requires no annotated data or training, only minimal human effort in evaluating and labeling 50 clusters for a corpus of almost 40 million tokens. Despite being unsupervised, our results show an accuracy that is comparable to the corpus annotations which result from an existing code switching algorithm and the combined usage of two supervised systems using character and byte n-gram models (Volk and Clematide, 2014).

1 Introduction

Language identification (LID) is important in NLP so long as the applications and tools designed and used are language-specific. Many tokenizers, POS-taggers, lemmatization and NER systems suffer in performance when met with sporadic sequences of foreign/unknown elements – this is especially the case when the languages in question are lesser known, the domain is more specific, and/or the training material is scarce. The view that LID is a "solved task" is unfortunately a misconception that is based on the success of work that dealt with document-level LID in a small number of languages (Lui, 2014). Real world data, esp. those with much code switching (CS), the phenomenon that occurs when speakers/writers switch back and forth between at least two languages in communication, on smaller spans of texts or with mixed genres, continue to pose challenges.

In this paper, we present our initial LID effort for a large, domain-specific, yet very diverse in themes and styles, multilingual diachronic corpus of periodicals published by the *Swiss Alpine Club* (SAC) abound with intrasentential CS instances. CS in this corpus has previously been addressed in Volk and Clematide (2014) who use a set of heuristics to identify code-switching candidates and label these using *langid.py*[1] (Lui and Baldwin, 2012), an off-the-shelf Python package providing a supervised multinomial Naive Bayes classifier that has been trained over a mixture of byte n-grams (1≤n≤ 4) on 97 languages. In contrast, we attempt to tackle this task through a simple cluster-and-label approach with unsupervised word vectors. Our system – requiring only minimal human intervention when labeling the induced clusters – achieves comparable performance to the existing CS annotations, demonstrating the feasibility of unsupervised word clustering for LID for our corpus.

2 Related Work

Word-level language identification has been tackled mostly through supervised approaches in prior work. For example, King and Abney (2013) show that the problem can be framed as a sequence labeling problem and that hidden Markov Model (HMM) and Conditional Random Fields (CRFs) can be trained – starting from monolingual data – to perform reasonably well at labeling words in multilingual texts.

[1] https://pypi.python.org/pypi/langid

Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP, pages 45–54,
San Diego, California, June 17, 2016. ©2016 Association for Computational Linguistics

In the first shared task on LID on code-switched data from 2014, system architectures ranged from a rule-based system to ones leveraging word embeddings, extended Markov Models, and CRF autoencoders (Solorio et al., 2014). While most teams focused on multilingual LID systems for the shared task, there are approaches that deal with classification on bilingual code-switched texts specifically – for example, Jhamtani et al. (2014) build a system that makes use of several heuristic features, including a special edit distance between Hindi and English that fits their use case for "Hinglish" texts.

On the other hand, unsupervised approaches to word-level language identification have not been as popular. A cluster-and-label approach similar to ours has recently been applied to unsupervised "Translationese" detection for machine translation by Rabinovich and Wintner (2015), but only for text passages of 2000 tokens each. Another characteristic that sets apart their approach from ours is the cluster labeling approach. Their automatic labeling approach builds "representative" language models for the class labels and then assigns them to the unsupervised clusters by comparing them to the empirical distributions in the clusters. Since their approach is not applicable in our case and labeling clusters for our task requires rather little effort, we resorted to manual labeling.

3 Data: SAC Yearbooks from Text+Berg (TB)

Text+Berg digital[2] is an ongoing text digitization and annotation project for alpine texts. The yearbooks published by the SAC form a substantial part of the corpus. They contain "reports and essays on all aspects of alpinism as well as alpine nature and culture" and cover a good range of subgenres: from more literary essays and anecdotal narratives on mountain expeditions, book reviews and poems, to practical travel tips such as hotel reviews and cabin directories, to more scientific studies of living organisms, glacier and climate observations, geo-historical descriptions on cols, mountains, and parks, and on the flora and fauna of the Alps and other mountain regions, to more technical accident and security reports and financial reports from protocols of the annual club gatherings (Volk and Clematide, 2014).

From the first edition of the yearbook in 1864 to 1923, the publication appeared yearly under the title *Jahrbuch des SAC*, and from 1925 until present, *Die Alpen. Les Alpes. Le Alpi*. Although it has been published monthly since 1996, the issues have been archived yearly as the yearbooks therebefore by TB – one XML file per year, and are referred to here as the yearbooks. Before 1957, these yearbooks are written without translation in mixed languages – German, French, Italian, and Romansch (as far as we know). That is, an article or a sentence in one language may contain passages or word(s) in one or many other language(s) without translation. Since 1957 there have been parallel versions of these yearbooks in German and in French, with the addition of Italian since 2012. But despite the emergence of these parallel versions, the occurrences of CS remains rather consistent throughout the years. Each of these parallel versions is archived as one XML file per year. For these 150 years between 1864 and 2014, there are 209 XML files (there was no publication in years 1870, 1915, and 1924), with roughly 39 million tokens in total after conversion into plain text, which we used as data for our study. The yearbooks from 1864 to 2000 were scanned and converted into text with commercial OCR software by *TB digital*. Through a crowdsourcing initiative, selected yearbooks have been manually inspected and almost all OCR errors from these books were corrected. The texts from 2001 to 2009 were extracted from PDFs. From 2010 on, *TB digital* has been directly receiving the publication in XML format from the SAC and converting these into the XML files with linguistic annotations available for download for research purposes.

Up to and including the latest release of the yearbooks (release 151v01, from April 11, 2015), all TB corpora have been LIDed on the sentence-level for all sentences with more than 40 characters using *Lingua-Ident*[3] by Michael Piotrowski, a statistical language identifier based on character n-gram frequencies. (Results for shorter sentences and for Romansch (RM) were found to be unreliable and

[2] http://textberg.ch

[3] http://search.cpan.org/dist/Lingua-Ident

these were hence assigned language tag of the previous sentence or that of the article). *Lingua-Ident* is able to recognize sentences in German (DE), French (FR), Italian (IT), and English (EN) thus far. Swiss-German (CH-DE) requires additional processing. While it is safe to narrow the number of dominant languages of the corpus down to 6 – the four official languages of Switzerland (DE, FR, IT, RM), in addition to EN and CH-DE – automatically determining exhaustively what other languages it contains will remain a hard problem. In our manual evaluation phase for the present experiment, we notice Latin (LA) and Spanish (ES) elements in the corpus, which confirm the observation by Volk and Clematide (2014), but we also found texts in Danish (DA) and Tibetan (BO) (in Roman script), for instance. As described in the following section, we approach our LID task as a closed-class classification problem (with 8 classes/labels) after clustering our word vectors.

4 Method

We adopt an unsupervised data-driven approach in clustering word vectors, optimized with lessons learned from Levy et al. (2015) by combining a PPMI (positive pointwise mutual information) matrix with a constant-sized weighted context window, after extracting plain text from each of the XML files for the SAC yearbooks. Our goal is to assign a label (i.e. a language class) to each word type in our data.

4.1 Word Vector Representation: PPMI matrix with weighted context window

We take as vocabulary V all word types v that neither are punctuations nor numbers, nor do they contain any. That is, tokens such as *1er* were excluded, regardless of whether it could be a permissible sequence in a language or an OCR error. We take as context words all elements of V that occur at least 100 times (for reasons of scalability), and refer to the set of these context words as C. We represent our data by constructing a high dimensional $|V| \times |C|$ matrix M with one row for each word type v and with one column for each context word c . This results in a 785,266 x 23,263 matrix.

We collect co-occurrence counts for every pair (v, c) using a context window of 5. Concretely, we treat each yearbook file as a sequence of tokens, $w_1, \ldots, w_N$, where N is the total number of these "eligible" (non-punctuation, non-numeric) tokens. For each w_i such that $w_i \in V$, we give a weighted count to each of the surrounding words, provided that they are in C: $\frac{1}{5}$ for w_{i-5} and w_{i+5}, $\frac{2}{5}$ for w_{i-4} and w_{i+4}, $\frac{3}{5}$ for w_{i-3} and w_{i+3}, $\frac{4}{5}$ for w_{i-2} and w_{i+2}, and 1 (i.e. $\frac{5}{5}$) for w_{i-1} and w_{i+1}. (Note that the context window is fixed, e.g. if only one of the surrounding words is in C, or if there is only one surrounding word, only one count will be collected, the window does not expand in order to collect counts 5 times.) We sum up these counts from all yearbooks.

The value of each matrix cell M_{jk} is the weighted co-occurrence count of (v_j, c_k) transformed into a normalized association measure, PPMI, where

$$PPMI(v, c) = \max(PMI(v, c), 0) \tag{1}$$

$$PMI(v, c) = \frac{\log P(v, c)}{P(v)P(c)} \tag{2}$$

and $P(v, c)$, $P(v)$, and $P(c)$ are estimated using maximum likelihood from the co-occurrence counts.

Pointwise mutual information (PMI), defined in eq. 2, is an association metric for measuring word association norms based on the information theoretic concept of mutual information. PMI compares the joint probability of two events with the product of their marginal probabilities.

Proposed by Church and Hanks (1990), the PMI of two events x, y is interpreted as follows:

> PMI(x, y) $\gg 0$ means that there is a genuine association between x and y, as P(x,y) will be much larger than P(x)P(y)
>
> PMI(x, y) ≈ 0 implies that there is no interesting relationship between x and y, as P(x,y) $\approx$ P(x)P(y)
>
> PMI(x, y) $\ll 0$ means that x and y are in complementary distribution, as P(x,y) will be much less than P(x)P(y).

Since there could be many entries in the PPMI matrix where (x,y) were never observed in the data, yielding $-\infty$ as PMI(x,y) = $\log 0$, we adopt the common practical solution of using PPMI, in which all negative PMI values are replaced by 0 (see eq. 1).

4.2 Dimensionality Reduction: Truncated Singular Value Decomposition (TSVD)

SVD factorizes a matrix M into the product of three matrices $U \cdot \Sigma \cdot V^T$, where U and V are orthonormal and Σ is a diagonal matrix of eigenvalues in decreasing order. When this factorized matrix is truncated, only the largest/top d diagonal elements of Σ are "kept". In other words, TSVD compresses the major associative patterns in the data into a lower dimensional matrix by ignoring the smaller, less important influences (Deerwester et al., 1990).

As mentioned in Section 4.1, our PPMI matrix has 23,263 columns, and despite its sparsity, the original word vector file is 37 GB in size. To facilitate our clustering experiments in the subsequent step, we reduce the dimensions of these word vectors to 100 using TSVD[4], resulting in a new file size of 1.21 GB (a reduction of nearly 97%).

4.3 Clustering: K-means

TSVD only gives us a matrix of reduced dimensions. But the number of words we have to cluster is still fairly high (785,266). K-means, as a flat, as opposed to hierarchical, clustering algorithm, is an efficient solution for our task. The objective of K-means is to minimize the average squared Euclidean distance of word vectors from their cluster centers where a cluster center is defined as the mean or centroid $\vec{\mu}$ of the word vectors in a cluster ω (Manning et al., 2008):

$$\vec{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\vec{v} \in \omega} \vec{v} \tag{3}$$

K in *K-means* refers to the number of clusters prespecified by the user. To obtain 50 clusters from our data, we first initialize 50 points (i.e. means) to random values, seeding the random generator with a random seed. The algorithm comprises two steps: the assignment step, in which each data point is assigned to the nearest centroid, giving rise to 50 clusters with one centroid for each cluster, and the update step, in which the centroids are then adjusted such that they represent the mean values of all the data points in the clusters formed in the assignment step. This is repeated until the cluster assignments converge.

In order to find a set of clusters that is most representative of our data, we run several clustering experiments varying the number of clusters. Ideally, we would get vocabulary items of a language to cluster together such that we get one cluster per language. As may be expected, the situation is a bit more complicated – e.g. if we divide up our data into 5 clusters, we get 1 mixed, 1 IT and 1 FR cluster, and 2 DE clusters. This tendency remains rather stable irrespective of the clustering algorithm used. In contrast, the number of clusters considerably impacts cluster quality. If too few clusters are used, only the strongest tendency of the data can be captured and minority languages such as RM and EN would be lumped together with the majority languages DE/FR/IT or contribute to mixed clusters. As we increase the number of clusters, purer clusters emerge and mixed classes become smaller. However, beyond a certain point, these improvements seem to level off – 100 clusters are not necessarily purer than 50 clusters. In earlier experiments, we compared both the scikit-learn implementation of a non-parametric Dirichlet Process Gaussian Mixture Model (DPGMM) which automatically determines the appropriate number of clusters from the data[5] and a parametric approach using K-means with 50 clusters. The non-parametric model found, depending on the choice of a spherical, diagonal, or full covariance matrix, 194, 201, and 4 clusters respectively. While we knew that 4 clusters would be too few, we measured performance of the other two against K-means (see Table 1) and decided to go with K-means for clustering as more clusters do not necessarily entail purer clusters or better results.

We found a set of 50 clusters (which are formed using the vectors we have been describing – these vectors differ from those in earlier experiments only in some minor variation in symbols filtered out in the process of vector building) we produced using K-means yields fairly reasonable language group-

[4] via the implementation in scikit-learn (Pedregosa et al., 2011): `http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html`

[5] One still has to provide an upper bound on the number of clusters for inference using scikit-learn. We chose 1,000.

	tsvd_dpgmm_diag (201 clusters)	tsvd_dpgmm_spherical (194 clusters)	tsvd_kmeans (50 clusters)
strict	88.78	88.60	88.88
lenient	89.39	89.30	89.55

Table 1: Accuracy scores (rounded to 2 digits) from non-parametric (Dirichlet Process Gaussian Mixture Model) vs. parametric models (K-means), using vectors built in earlier experiments

ings, as even minority languages such as RM and EN could be represented as classes of their own. We hence use it for evaluation of our TBLID system in this paper.

4.4 Evaluating and Labeling of Clusters

We evaluate the clusters manually and assign labels (language classes) in a majority rule fashion. If the first 20 most frequent words in a cluster look to be from one language, that language is assigned as one of the class labels: DE, EN, FR, IT, RM, CH-DE, NE (named entity), or MIXED (if words are not in any of the aforementioned classes). (See Figure 1 for examples of clusters that were labeled DE, FR, IT, EN, and RM.) Most of the time, there is a majority class in a cluster. In cases where it was difficult to make a call based on the top 20 words, the remainder of the cluster will be looked at and evaluated.

The need for a separate class for NEs is based on cluster results[6]. Although certain NEs have variants in different languages, e.g. the Swiss city of Basel is *Basel* in DE, *Bâle* in FR, *Basilea* in IT, and these words could fall into their respective language classes, results of various clustering experiments show that, especially when the number of classes are well above the number of language classes, some NEs do tend to cluster together.

5 Evaluation of TBLID and Results

5.1 Evaluation Setup

With one language class label assigned for each vocabulary item, our experiment to evaluate our LID system (henceforth: TBLID) begins by randomly selecting sentences that are indicated to contain CS segments based on the annotations that are about to be included in the upcoming release of the yearbooks provided by *TB digital*. These annotations result from the implementation of the CS detection algorithm from Volk and Clematide (2014) (hereafter: VCCS) which uses a combination of four factors (the presence of quotation marks, at least 2 tokens being outside of these quotation marks, lemma tags ⟨unknown⟩, and minimum CS segment length of 15 characters) as cues to identify CS instances. According to these criteria, 194 of the 209 yearbook files contain at least one CS sentence. We randomly select one CS sentence from each of these 194 files to evaluate TBLID based on word-level language label accuracy and to compare our labels to the TB annotations which are output of *langid.py* for the intrasentential CS segments and *Lingua-Ident* on the sentence level. (Classifying words one by one using *langid.py* alone for these 194 sentences yielded an overall accuracy of approximately 36%, hence it will not be used for comparison in this study.)

Word-level language identification accuracy is the percentage of the correctly labeled word tokens (i.e. tokens containing no numeric element or punctuation except for hyphen(s), apostrophe(s), and period(s)) from all sentences with CS. 2 of these 194 sentences are disqualified for evaluation due to indecipherability as they consist exclusively of numeric elements and abbreviations, most of which are potential OCR errors. This leaves us with a total of 5,073 word tokens.

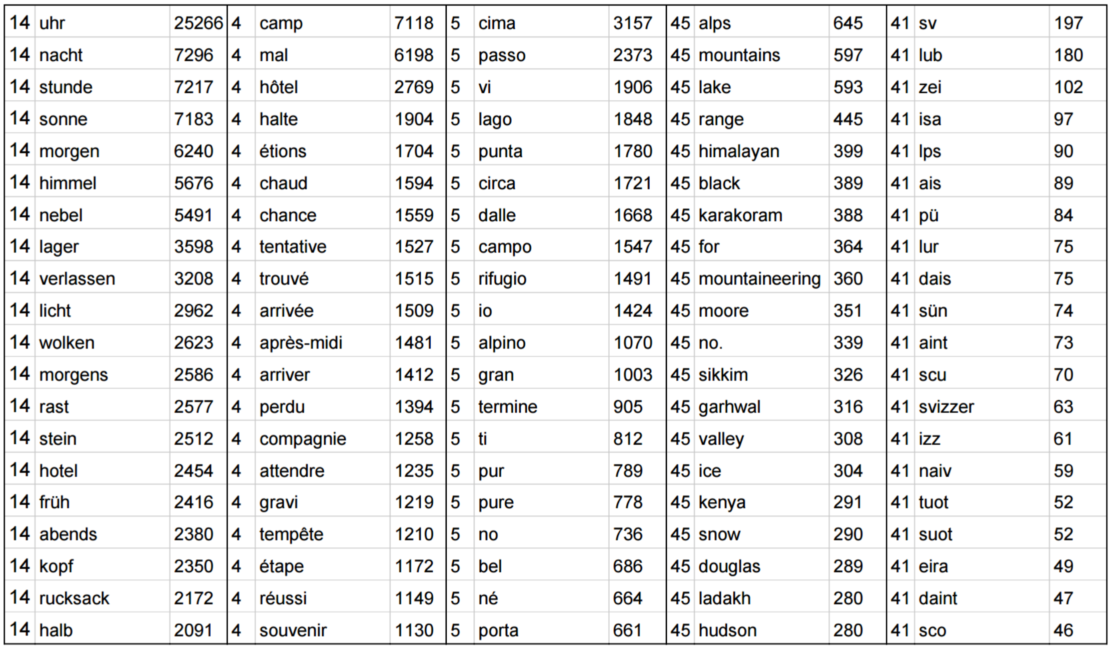
We illustrate scoring here with the following sentence from the 1925 yearbook:

> [1] Je n' en sais rien , mais l' énergie de son «
> Oel per oel e daint per daint » résonne encore
> à mon oreille .

The 22 word tokens here should be identified with the following languages (labels indicated here with preceding underscores, CS segment boldfaced):

> [1a] **GOLD**: Je_fr n'_fr en_fr sais_fr rien_fr
> , mais_fr l'_fr énergie_fr de_fr son_fr «

[6] but this also concurs with the annotation guidelines for the shared task in EMNLP 2014 Workshop on Computational Approaches to Code Switching: `http://emnlp2014.org/workshops/CodeSwitch/guideline/word_annotations_en_es_CF.pdf`



14	uhr	25266	4	camp	7118	5	cima	3157	45	alps	645	41	sv	197
14	nacht	7296	4	mal	6198	5	passo	2373	45	mountains	597	41	lub	180
14	stunde	7217	4	hôtel	2769	5	vi	1906	45	lake	593	41	zei	102
14	sonne	7183	4	halte	1904	5	lago	1848	45	range	445	41	isa	97
14	morgen	6240	4	étions	1704	5	punta	1780	45	himalayan	399	41	lps	90
14	himmel	5676	4	chaud	1594	5	circa	1721	45	black	389	41	ais	89
14	nebel	5491	4	chance	1559	5	dalle	1668	45	karakoram	388	41	pü	84
14	lager	3598	4	tentative	1527	5	campo	1547	45	for	364	41	lur	75
14	verlassen	3208	4	trouvé	1515	5	rifugio	1491	45	mountaineering	360	41	dais	75
14	licht	2962	4	arrivée	1509	5	io	1424	45	moore	351	41	sün	74
14	wolken	2623	4	après-midi	1481	5	alpino	1070	45	no.	339	41	aint	73
14	morgens	2586	4	arriver	1412	5	gran	1003	45	sikkim	326	41	scu	70
14	rast	2577	4	perdu	1394	5	termine	905	45	garhwal	316	41	svizzer	63
14	stein	2512	4	compagnie	1258	5	ti	812	45	valley	308	41	izz	61
14	hotel	2454	4	attendre	1235	5	pur	789	45	ice	304	41	naiv	59
14	früh	2416	4	gravi	1219	5	pure	778	45	kenya	291	41	tuot	52
14	abends	2380	4	tempête	1210	5	no	736	45	snow	290	41	suot	52
14	kopf	2350	4	étape	1172	5	bel	686	45	douglas	289	41	eira	49
14	rucksack	2172	4	réussi	1149	5	né	664	45	ladakh	280	41	daint	47
14	halb	2091	4	souvenir	1130	5	porta	661	45	hudson	280	41	sco	46

Figure 1: Examples of clusters labeled DE, FR, IT, EN, and RM (from left to right). The leftmost column of the three columns for each language (i.e. inside each frame) indicates the cluster number, the middle column the word token, the rightmost the term frequency.

Oel_rm per_rm oel_rm e_rm daint_rm per_rm daint_rm » résonne_fr encore_fr à_fr mon_fr oreille_fr .

Annotations by *TB digital* (punctuation marks are also classified by their system, but excluded from evaluation for this paper) have 15 of these correctly identified as FR and 7 RM words incorrectly identified as OC (Occitan):

[1b] **TB (Lingua-Ident + langid)**: Je_fr n'_fr en_fr sais_fr rien_fr ,_fr mais_fr l'_fr énergie_fr de_fr son_fr «_oc **Oel_oc per_oc oel_oc e_oc daint_oc per_oc daint_oc** »_oc résonne_fr encore_fr à_fr mon_fr oreille_fr ._fr

TBLID has 17 correctly identified word tokens and makes 5 mistakes (non-word tokens were excluded as mentioned, and given *_unk* tags):

[1c] **TBLID**: Je_fr n'_fr en_fr sais_fr rien_fr ,_unk mais_fr l'_fr énergie_fr de_fr son_fr «_unk **Oel_mixed per_it oel_mixed e_it daint_rm per_it daint_rm** »_unk résonne_fr encore_fr à_fr mon_fr oreille_fr ._unk

If *oel* were a word in a language not available in the set of language classes in TBLID, classifying it as MIXED would be acceptable and expected, but since RM is a possible label, each occurrence of *oel*, for example, is counted as 1 incorrect instance.

All word tokens are evaluated in context, in the particular instance, not for their potentiality to take on a certain label. E.g. the two word tokens *par an* meaning *yearly* in a FR segment should both be labeled FR, even though *par* can be DE/EN/FR/IT/RM, and *an* can be an EN, FR, DE, as well as a RM word, if evaluated independently. We will return to the issue of multilingual homographs in our discussion in Section 6.1 below.

5.2 NE Classification and Two Annotation Schemes

On the one hand, it can be non-trivial to pinpoint what kinds of NEs the clusters are supposed to classify when it is more so the case that certain "proper-noun-looking" nouns tend to form their own clusters; on the other hand, we do see cases that make "human sense", such as names of months of a language clustering together. For the purpose of this

LID task, we define organizations, person names, locations (which include mountain and cabin names), and month names as NEs.

Since NE is not an available category in TB annotations, we consider an NE correct if an LID system gives it an NE or a correct language label. But this notion of "correct language" can be an intricate matter with such an inherently code-switched corpus. We may understand that, for example, within a DE segment, if the word *Bâle* was used when there is a valid DE variant *Basel* available, there is valid reason to consider this as a CS element. But how about in cases where the form of the name is identical across the languages in question, such as *Bonn*? The usage of *Bonn* in midst of a FR segment may not be considered CS in that case. But the line here, if there is one, can be blurry. Consider the following sentence from the 1975 German yearbook:

> [2] Indessen gibt dann der Guide bleu von 1962 einige Details über die Bedingungen für eine Besteigung bekannt , indem er von der alten Ausgabe die Erwähnung warmer Kleidung , guter Schuhe und Schutzbrillen übernimmt , aber alles in Verbindung mit «Grand Hotel Ätna , mehreren Restaurants bei der Casa Cantoniera , Schutzhütte Sapienza , Hütte Menza des CAI , Wintersport etc . »

If annotated with the above rationale (sentence truncated to highlight only the more relevant parts):

> [2a] **GOLD-strict:** Indessen_de gibt_de dann_de der_de **Guide_de** **bleu_de** von_de 1962_unk einige_de Details_de über_de die_de Bedingungen_de für_de eine_de Besteigung_de bekannt_de , . . . , mehreren_de Restaurants_de bei_de der_de **Casa_de_ne** **Cantoniera_de_ne** , Schutzhütte_ne_de **Sapienza_ne_de** , Hütte_de_ne **Menza_de_ne** des_de **CAI_ne_de** , Wintersport_de etc_de . »

Classifying *Guide bleu*, *Casa Cantoniera*, *Sapienza*, *Menza*, and *CAI* (which stands for *Club alpino Italiano*) as DE may not be uncontested, perhaps even a bit unintuitive for some. We hence devised two evaluation schemes – strict and lenient. The annotations as exemplified in [2a] are considered strict. The lenient annotations are illustrated in [2b] below:

	Lingua-Ident + langid.py	TBLID
strict	89.73	89.33
lenient	89.81	89.91

Table 2: Final accuracy scores (rounded to 2 digits), based on 5073 words

> [2b] **GOLD-lenient:** Indessen_de gibt_de dann_de der_de **Guide_de_fr** **bleu_de_fr** von_de 1962_unk einige_de Details_de über_de die_de Bedingungen_de für_de eine_de Besteigung_de bekannt_de , . . . , mehreren_de Restaurants_de bei_de der_de **Casa_de_ne_it** **Cantoniera_de_ne_it** , Schutzhütte_ne_de **Sapienza_ne_de_it** , Hütte_de_ne **Menza_de_ne_it** des_de **CAI_ne_de_it** , Wintersport_de etc_de . »

There can be more than one permissible language for each word. For the word *Casa* here, for example, a label is considered correct if it is DE, NE, or IT. The strict annotations for a word are a subset of the lenient annotations.

5.3 Results

Despite being a very weakly supervised system (manual "supervision" took place only in the labeling of 50 clusters), the combination of the two fully supervised systems and TBLID performed neck and neck. As expected, the strict annotation standard favors the combination of *Lingua-Ident* and *langid.py*, which assumes the notion of a base language for each chunk of texts within a sentence. TBLID is word-level and allows for more flexible switching and more variation in languages within a sentence, hence agrees more with the lenient annotation standard. In the strict evaluation, TB annotations have 4552 out of 5073 word tokens correct, with an accuracy score of 89.73%, while TBLID has 4532 correct, at 89.33%. In the lenient evaluation, TBLID has 4561 correct, i.e. accuracy of 89.91%, whereas TB annotations only have 4556 correct, at 89.81% (all scores are rounded to 2 decimal places), as summarized in Table 2.

6 Error Analysis and Future Directions

6.1 Multilingual Homographs

The issue that came first to our attention was effected by what we term "multilingual homographs" (MHs) and the system of assigning one label per word.

Generally, a homograph (within one language) is a word form that expresses two or more different meanings, e.g. *bass* in EN can refer to a fish and a musical instrument. In our study, we use MH to refer to a word form that exists in two or more languages. TBLID assigns one label to each word type, that is, word type of a multilingual corpus. This immediately poses a problem, above all, to the scores of stopwords that the few Indo-European languages in this corpus have in common. Word forms such as *la*, *il*, *le*, *de*, *da*, *des*, *d'*, *an*, *in*, *on*, *per*, *si*, *non*, *et*, *i*, *was*, *a*, *je* as well as *aller*, *va*, *est*, *qua*, *be*, *god*[7], *ni*, *sur*, *at*, *mal*, *termine* are eligible word forms in more than one of the languages relevant in our LID task. Close to about 100 instances where TBLID "mislabeled" in this evaluation were of errors related to MHs. For example, the TBLID label for the word *des* (meaning: "of the" in FR and DE) is FR. Hence whenever *des* in DE appears, our current "one label per word type"-design suffers an inevitable defeat (and a frequent one too as *des* occurs not at all seldom in DE). We tried to remedy this situation by attempting to assign multiple labels via soft clustering with EM (Expectation Maximization) and TSVD via scikit-learn. But the distribution of these MHs are so skewed that it was difficult to even pick a threshold above which the word should be classified as belonging to a certain cluster. Most of these popular MHs, despite their multiple identities, (oddly) show a strong preference for one cluster with a probability of almost 1.

Since it is less likely for function words to be an independent CS element, modeling context through a sequential model such as HMM and CRF as in King and Abney (2013) after our clustering effort could be helpful. Instances such as the one in Section 6.2 below are good test cases for these future experiments. They serve as vivid reminders that relying on the LID of a larger stretch of text for a base language can also be prone to missing many cases of CS and correct LID.

6.2 CS Detection Heuristic: Quotation Marks

One frequent error pattern for TB annotations is the assumption of quotation marks playing a role in CS. The following sentence from the 1997 FR yearbook contains a long stretch of pairs of words in the form "FR / DE" with figures reporting the financial gains, losses, and interests of the SAC fund. The sentence was extracted as a CS candidate for evaluation due to the « and » towards the end of the sentence:

> [3] Franz Werthmüller , chef des finances … Zins / Intérêts … Material Rettungswesen / Matériel de sauvetage … Veränderung Clubvermögen / Variation de la fortune du club … Zunahme / Augmentation … Abnahme / Diminution Zu- und Abgänge FondsAttributions et débits des fonds Verzinsung der Fonds / Intérêts des fonds Zunahme Allgemeine Reserven / Augmentation Réserves générales … Details siehe Tabelle « Veränderung der Fonds » / Détails :

VCCS assumes elements outside of quotation marks to be in one (base/default) language, merely that identified on the sentence-level by *Lingua-Ident*. According to this, all word tokens of this sentence were predicted to be FR, except for the 3 inside the « and » , which *langid.py* classified correctly as DE (sentence accuracy: 50/78). TBLID, on the other hand, identified more words correctly – getting DE for *Zins* and FR for *Intérêts* as "interest" for the two respective languages. It had 65 out of 78 correct and even recognized *Franz* and *Werthmüller* as NEs.

Another motivation that calls for a refinement of VCCS (from 1877 yearbook):

> [4] **GOLD-lenient:** in_de Vissoye_de_ne_fr erfuhr_de ich_de von_de einem_de Thalkundigen_de , sie_de hätten_de « **mauvaises_fr femmes_fr** » zum_de Schlupfwinkel_de gedient_de , und_de da_de ich_de weiter_de forschte_de , was_de denn_de diese_de « **mauvaises_fr femmes_fr** » verübt_de , erhielt_de ich_de die_de zögernde_de Erklärung_de , es_de seien_de « **sorcières_fr** » gewesen_de .

This sentence was extracted as a possible CS candidate due to their *« mauvaises femmes »* segments.

[7] meaning "forest" or "wood" in RM, as per `http://www.pledari.ch/mypledari/index.php`

But VCCS does not handle the segment *«sorcières»*, as it is shorter than 15 characters in length, letting it default into the base language of the sentence (DE). TBLID was able to tag the word *sorcières* with FR. Here we see the feasibility of a word-level LID system. (Score information: both systems get 33 out of 34 correct – TB annotations miss on the word *sorcières*, while TBLID recognizes *Thalkundigen* (a DE compound noun meaning "those who are knowledgeable about the valley") as an NE.)

6.3 OCR errors

Despite the crowdsourcing effort to rectify OCR errors for some yearbooks, OCR errors are still bountiful in the corpus, some of them even form clusters of their own. Our approach could be refined through iterating the clustering procedure, mirroring the two-phase setup in Rabinovich and Wintner (2015)) in which the unsupervised clustering algorithm was run twice – separating genre in the first, and Translationese and Original in the second run. This staged approach might help clean up some of our mixed clusters if we, for example, remove the purer clusters (including these "OCR clusters") from the sample first and then re-cluster the remaining words.

6.4 Contextual Accuracy: Diachronic Corpus

Consider the following sentence snippet from the 1874 yearbook:

> [5] **TBLID:** . . . von_de Bünden_de an_de den_de Wiener_de Congress_en und_de nach_de Mailand_de . . .

EN is a minority language in this corpus. Mislabeled EN words are plentiful in our study as EN data is sparse. The system was, however, clever enough to classify *Congress* as EN, which would have been correct were it not for the fact that the word here is the DE word *Kongress* with an archaic spelling.

7 Conclusion

We have presented a data-driven, self-sufficient, cluster-and-label, simple count distributional approach that identifies the language of word types of a multilingual domain-specific corpus of almost 40 million tokens. We report an accuracy of 89.91% based on about 5,000 word tokens evaluated and the only "supervision" required was the labor to manually label 50 clusters. We noted some fundamental issues in the definition of gold standard in LID and devised two annotation standards. Different arguments speak for and against various possibilities to optimize the system – the assignment of multiple labels (esp. for multilingual homographs), a more explicit but smart modeling of surrounding context (for example, through combining clustering with HMM model after clustering, or staging clustering itself), and the refinement of the code switching detection algorithm proposed by Volk and Clematide (2014). Through our comparison with off-the-shelf alternatives, we learned that LID of small(er) segments is far from a solved task. Future directions may also include investigating whether incorporating supervised methods would improve performance and testing this method on other datasets.

Acknowledgments

We would like to thank *Text+Berg digital* for sharing their existing code-switching annotations and helpful suggestions. We would also like to thank the three anonymous reviewers for their valuable comments.

References

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407.

Harsh Jhamtani, Suleep Kumar Bhogi, and Vaskar Raychoudhury. 2014. Word-level language identification in bi-lingual code-switched texts. In Wirote Aroonmanakun, Prachya Boonkwan, and Thepchai Supnithi, editors, *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation, PACLIC 28, Cape Panwa Hotel, Phuket, Thailand, December 12-14, 2014*, pages 348–357. The PACLIC 28 Organizing Committee and PACLIC Steering Committee / ACL / Department of Linguistics, Faculty of Arts, Chulalongkorn University.

Ben King and Steven P. Abney. 2013. Labeling the languages of words in mixed-language documents using weakly supervised methods. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 1110–1119. The Association for Computational Linguistics.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 25–30. The Association for Computer Linguistics.

Marco Lui. 2014. *Generalized language identification*. PhD thesis, The University of Melbourne.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Ella Rabinovich and Shuly Wintner. 2015. Unsupervised identification of translationese. *TACL*, 3:419–432.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar, October. Association for Computational Linguistics.

Martin Volk and Simon Clematide. 2014. Detecting code-switching in a multilingual alpine heritage corpus. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 24–33, Doha, Qatar, October. Association for Computational Linguistics.