rijksuniversiteit groningen

faculteit der letteren

RIJKSUNIVERSITEIT GRONINGEN

Automatic Classification of English Learner Proficiency Using Elicited Versus Spontaneous Data

by

Xiaoyu Bai

First Supervisor: Dr. Malvina Nissim (Rijksuniversiteit Groningen)
Second Supervisor: Dr. Albert Gatt (University of Malta)

A thesis submitted in partial fulfillment for the degree of
ReMa Linguistics
in the
Faculty of Arts

August 2018

RIJKSUNIVERSITEIT GRONINGEN

Abstract

Faculty of Arts

ReMa Linguistics

by Xiaoyu Bai

We present a novel study on CEFR level prediction in writings by non-native speakers of English, using on the one hand clean data elicited in a language learning context, and on the other hand noisy, spontaneous data from social media. The elicited data were drawn from the freely accessible learner corpus *EF-Cambridge Open Language Database* and consist of level-matched short essays written as part of an online English course. The spontaneous data were gathered from the social media platforms Twitter and Reddit, where users' self-reported proficiency levels were used as distant labels. Our level classification experiments were run both *within* and *across* the two domains as well on *mixed-domain* data. They were mainly conducted using linear SVM and logistic regression, although we also briefly explored bidirectional LSTM and convolutional neural networks, particularly in a setting of multi-task learning. We find that distant supervision based on user self-reports is a viable option of automatically generating noisily labelled training data for learner level prediction from social media texts. Despite the noisy nature of both the data and the labels, level prediction within the social media domain proved to be feasible, with system performance clearly beating the majority class baseline. Classification across the two domains, however, was revealed to be unsuccessful.

Keywords: Learner level classification, NLP, social media, machine learning

Acknowledgements

First of all, I would like to express my deepest gratitude to my main supervisor, Dr. Malvina Nissim, who not only took much time to offer me valuable advice, point me to literature and resources and to give me insightful feedback on my results in our weekly meetings (which often ended up occupying her lunch break), but also always calmed my worries about the project with her cheerful smiles and assurances. I always greatly enjoyed our discussions and chats. Moreover, I would also like to thank my second supervisor Dr. Albert Gatt for his helpful feedback and suggestions during our Skype meetings. Furthermore, I would like to thank Angelo Basile for taking time to help me with the implementations of my neural models despite being busy with his own thesis.

During this project, I was very fortunate to have the friendship of Nina H. Kivanani, Claudia Zaghi, Elizaveta Kuzmenko, Anastasia Serebryannikova, Micha de Rijk and others, who motivated and supported me with companionship and fun conversations. I also want to thank Aina Gari-Soler and Carlos Diez-Sanchez for being great office mates in Malta and for encouraging me when I was having my anxieties about transitioning from theoretical to computational linguistics during this programme.

I also owe my thanks to the LCT coordinators for their help with any administrative problems I had during these two years. Moreover, I am indebted to the German *Studienstiftung des deutschen Volkes* and the *Erasmus+* programme for their financial support, which I highly appreciate.

My thanks also go to the teachers and fellow students of the *Mary Jane Bellia Ballet School* in Malta, the *Wanda Kuiper Ballet School* in Groningen as well as the *USVA* ballet class at University of Groningen for always giving me something unrelated to academia to look forward to during my studies in these past two years.

Finally, I am deeply thankful to my parents and my friends Lisa Seelau in Germany and Laura Golin in Italy for their constant moral support when I met challenges in my studies. Without their encouragement, I would certainly have had more struggles during this study programme and this thesis project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Using Natural Language Processing (NLP) technologies for applications related to second language teaching and learning has been a field of great interest. For instance, second language (L2) readability assessment (Crossley, Greenfield, & McNamara, 2008; Xia, Kochmar, & Briscoe, 2016) focuses on *input* material to learners, i.e. texts written by native speakers *intended* for L2 learners, and attempts to automatically recognise how "readable" a given piece of text is in terms of difficulty and complexity, usually predicting what proficiency level an L2 learner should be at in order for the text to make suitable reading material. Furthermore, a series of L2-related NLP applications focus on L2 users' *output* material, i.e. texts written by non-native speakers. For instance, Hawkins and Buttery (2010) conduct corpus-based analyses of learner language; automatic error correction (Leacock, Chodorow, Gamon, & Tetreault, 2010; Ng et al., 2014; Bryant & Ng, 2015) tackles the difficult task of recognising and correcting grammatical errors in learner-generated texts.

One task which focuses on learner output texts and has been of particular academic and commercial interest has been the automated assessment and grading of texts written by learners in language exams (Attali & Burstein, 2006; Yannakoudakis, Briscoe, & Medlock, 2011). Related to that is the automatic prediction of learners' proficiency levels in terms of a pre-defined scale or levels, based on their writings (Vajjala & Loo, 2014; Tack, François, Roekhaut, & Fairon, 2017). In this thesis project we explore English learner level prediction based mainly on writings on social media platforms.

1.1 Project Task Definition and Research Questions

Elicited Data When it comes to the automated prediction of learner levels, we notice that the focus is mostly (if not exclusively) on learner texts composed as part of some standardised exam or placement test (Vajjala & Loo, 2014; Hancke & Meurers, 2013; Tack et al., 2017). We consider this type of data to be *elicited* data, in that they do not arise from natural communication, but are produced "on cue" as part of a writing task, mostly in response to specific or even topic-defining prompts. In all likelihood, the non-native speaker will have paid special attention to writing well. There are clearly practical reasons for using elicited data for this task: Most important of all, data from the exam context are likelier to come with gold-standard level labels, produced by human examiners.

Spontaneous Data However, presumably, when a non-native speaker achieves a certain proficiency level, that level should be reflected not only in the controlled exam context, but also in natural, spontaneous and "random" language production, for instance, when they write in web forums or comment on posts on social media. Here, they do not write in response to someone else who wishes to obtain writings from them for the purpose of assessing their proficiency levels. We consider this kind of data *spontaneous* data.

In the project described in this thesis, we look at automatically predicting the proficiency level of non-native English speakers using elicited data on the one hand and spontaneous data on the other hand, with a focus on the latter. We will use the proficiency scale proposed by the *Common European Framework of Reference for Languages* (CEFR) (Council of Europe, 2001). The spontaneous data will be obtained from social media. Our core research questions are the following:

1. Can we use distant supervision to obtain a set of level-annotated, spontaneous data from social media in order to perform level prediction with supervised machine learning methods?
2. Is it at all possible to predict non-native speakers' proficiency level based on their writings in the noisy social media domain?
3. Given that most data with reliable level-annotation does come from the language exam context, can the language exam domain inform the social media domain?

1.2 Organisation of Present Thesis

We provide background information pertaining to the present project in Chapter 2, in which we address both the fields of NLP for learner assessment as well as NLP applications on social media data, as both fields are highly relevant to our project. Chapter 3 illustrates our data collection process: We describe our method of harvesting spontaneous data from the social media platforms Twitter and Reddit, based on distant supervision, and the learner corpus from which we draw our set of elicited data. Chapter 4 details our classification experiments using two traditional machine learning algorithms, the Support Vector Machine (SVM) and the logistic regression model. We discuss our methods and results and also briefly explore an innovative method of evaluation which might better suit the present task than the traditional evaluation metrics for multi-class classification. In Chapter 5, we look at a pilot experiment which applies two neural systems in a multi-task learning setting to the level prediction task. Finally, Chapter 6 concludes the present thesis and offers a few outlooks for future research in this field.

Chapter 2

Background

2.1 CEFR Learner Proficiency Levels

One of the most widely known and accepted scale for describing and measuring the proficiency level of a foreign language learner is the six-level proficiency scale by the *Common European Framework of Reference for Languages* (CEFR) (Council of Europe, 2001). The CEFR-system distinguishes between the six proficiency levels shown in Table 2.1, which fall into the three more coarse-grained classes of *basic*, *independent*, and *proficient* users.

A1	A2	B1	B2	C1	C2
Basic User		Independent User		Proficient User	

Table 2.1: The six foreign language proficiency levels in the CEFR system

The Council of Europe provides rough descriptions concerning what learners at each of the six levels "can do". To name a few examples, this ranges from "Can understand and use familiar everyday expressions and very basic phrases [...]" at level A1 to "Can express him/herself spontaneously, very fluently and precisely, differentiating finer shades of meaning even in more complex situations" at level C2. The CEFR proficiency levels have been used in several level prediction studies summarised in the section below. They will also be used in our own project.

2.2 Automated Essay Scoring and Level Prediction

The focus in the automatic prediction of learner levels and the highly related task of automatically scoring the essays in foreign language exams lies in the writings produced by language learners. It is assumed that properties in their written production will provide clues by which to classify or grade their levels of proficiency.

Automated Essay Scoring The automatic grading of free text writings in standardised foreign language exams can be considered a special case of automatic essay scoring, applied to learner essays and presumably with a greater focus on linguistic soundness. The final system output tends to be a score on a continuous scale. Obviously, given the amount of standardised English tests taken worldwide, including TOEFL, IELTS and Cambridge's English as a Second or Other Language (ESOL) exams, practical and commercial motivations for automatic marking of such exam texts are plentiful.

One of the earlier but well-known commercial systems is the Education and Testing Service's (ETS) *e-Rater* (Burstein et al., 1998; Burstein, 2003; Attali & Burstein, 2006). It uses a set of hand-designed numerical features which target individual, high-level aspects of learners' essay, such as grammar and language usage, organisation and prompt-specific vocabulary usage. Amongst the features targeting grammar, for instance, is the total count of (automatically detectable) language errors, normalised by the length of the essay. Based on these features, they derive a component score for each of these aspects, and the full essay score is then computed by combining these component scores, e.g. by taking their weighted average, which is the policy adopted in the second version of e-Rater (Attali & Burstein, 2006). Finally, a linear transformation can be applied to the resulting score to obtain an interpretable score on the desired marking scale.

In contrast, Yannakoudakis et al. (2011) advocate a preference ranking approach to assessing Cambridge ESOL exam scripts, showing its superiority over a regression model using the same features. They use data from the Cambridge Learner Corpus (Nicholls, 2003), which contains scripts composed by non-native English speakers taking the Cambridge ESOL exams. Based on a selection of linguistic features, including word n-grams, part-of-speech (POS) n-grams, the length of the script and the rate of errors etc., they let a binary SVM learn the correct *rank* which holds amongst the training samples by learning the rank between each pair of scripts. It thus learns the relative mark of each individual script with respect to all other scripts. The training inputs to the SVM are difference vectors representing the difference between (the feature representations of) a given pair of samples. In the test phase, the trained model outputs a rank amongst the test scripts, which can then be mapped to any marking scheme. The authors evaluate their system by measuring the correlation between their system's output and the true scores recorded in CLC as well as with four further scores given by senior human examiners. They find that their system achieves an average Spearman's rank correlation score of 0.721, which is only around 5 points below the upper-bound of human-human agreement.

Automated Level Prediction A few studies have looked at classification of learner essays into discrete classes of proficiency levels, including the CEFR levels, and the task has been applied to several languages. With respect to English, Crossley, Salsbury, and McNamara (2012) investigate level prediction of learner texts based solely on *lexical* competence, placing special emphasis on identifying the most predictive features which relate to different measures of non-native speakers' lexical competence. Such measures include, amongst others, their lexical diversity, their understanding of polysemous and homonymous words, the number of associations they make with given words etc. For their study, Crossley et al. (2012) obtain open-topic written essays from 100 learners of English of three proficiency levels. Using a Discriminant Function Analysis, they find that amongst the top predictive features are the diversity of learners' lexicon and some properties of the words they use, including the words' imagability, frequency and familiarity. Based on these discriminative features, the authors achieve an F1-measure of around 70%. Notice that the data used in this study are open-topic and have not been composed in response to specific, topic-related questions. Nonetheless, they have still been elicited and produced in a controlled context.

Vajjala and Loo (2014) examine automatic level prediction on essay texts written by learners of Estonian, using four out of the six CEFR levels (due to lack of data for the two levels excluded). They use a range of linguistic features and find morphological features and features related to lexical variety to be particularly predictive. This is unsurprising as Estonian is a strongly agglutinative language characterised by complex morphology. Their best model, an SVM, reports an accuracy of 79%.

Similarly, Hancke and Meurers (2013) apply CEFR-based level prediction to texts written by learners of German. Their data are drawn from the German portion of the MERLIN corpus (Boyd et al., 2014), a collection of CEFR-labelled learner essays in German, Italian and Czech. Drawing on insights from Crossley et al. (2012), they use a series of linguistic features such as lexical diversity and word frequency features, but also morphological and syntactic features such as the number of clauses per sentence and the depth of a sentence's constituency parse tree. They use five of the six CEFR levels, excluding only the highest C2 class. With a linear SVM, they achieve classification accuracy scores of 62%.

Level Prediction on Short Samples All of the above studies focus on essays, i.e. written samples of a certain length. Tack et al. (2017) are amongst the first that we are aware of to attempt classifying learners' proficiency to CEFR levels based on *short* answers, where samples range from 30 to 150 words. In their experimental setting, non-native speakers at different CEFR levels are prompted to provide short answers to open questions. Based on these replies, they were assessed and assigned to a gold-standard proficiency class via majority vote amongst three certified Cambridge examiners. While the human examiners have reported difficulties in assessing very short samples, the author report no correlation between the length of the samples and the level of disagreement between the three examiners. The system designed to automatically classify the learners' level based on these short answer scripts uses a soft-voting ensemble classifier, which consists of five traditional machine learning algorithms: Naive Bayes, decision tree, k-nearest neighbour, logistic regression, and SVM with a polynomial kernal. The classification features used are similar to the ones mentioned above. Amongst others, lexical diversity, average sentence and word length features have been shown to be particularly predictive. Classifiyng into five CEFR-levels, with levels C1 and C2 conflated into one class, they obtain an accuracy score of 53% and a macro-average F1-measure of 49.5%.

Notice that, in all of the studies reported here, learners' written production has been explicitly prompted with certain questions or tasks, generally in the context of an exam. Their setting is such that the learners write with the awareness that their writings will be recorded for the purpose of assessing their proficiency. Thus, all of these studies are concerned with what we consider *elicited* data. As mentioned, we wish to examine CEFR-based level prediction on *naturally* and *spontaneously* produced data, which we plan to harvest from social media.

2.3 NLP on Social Media Data

In recent years, with the rising popularity of various social media platforms, NLP technology has been increasingly applied to the social media domain for a range of tasks. Platforms such as Twitter, Reddit and Facebook offer a source of abundant and easily accessible (albeit also noisy and unannotated) data. Special tools have been developed to deal with the idiosyncrasies of social media and web language, such as the uncanonical use of spellings and punctuation marks, the usage of emoticons and special characters etc. The Python NLP module SpaCy (Honnibal & Montani, 2017) has a model for English which takes the web genre into account[1]; Carnegie Mellon University's NLP group provides a series of tools for application to Twitter data[2], e.g. a tweet parser (Gimpel

[1] https://spacy.io/models/en#en_core_web_sm

[2] http://www.cs.cmu.edu/~ark/TweetNLP/

et al., 2011). To name just a few examples, the following are tasks in which NLP is applied to data from social media, especially from Twitter:

Author Profiling Since 2013, there has been an annual shared task on (multilingual) author profiling at PAN (Rangel, Rosso, Koppel, Stamatatos, & Inches, 2013; Rangel et al., 2014; Rangel, Rosso, Potthast, Stein, & Daelemans, 2015; Rangel et al., 2016; Rangel, Rosso, Potthast, & Stein, 2017; Rangel, Rosso, Montes-y Gómez, Potthast, & Stein, 2018). Based on a person's writings on Twitter, blogs, and other social network platforms, the goal is to classify the person to a pre-specified age, gender and/or native language group. In other words, NLP systems attempt to extract information on a user's profile, based on his/her writings on the web. In the most recent 2017 and 2018 editions, the traditional SVM with n-gram features and recurrent neural networks such as the Bi-LSTM (Hochreiter & Schmidhuber, 1997) have proved to perform particularly well (Basile et al., 2017; Rangel et al., 2017, 2018).

Sentiment Analysis Another task of especially high commercial relevance is sentiment analysis (or opinion mining) on social media, which has also formed the topic of several shared tasks (Nakov, Ritter, Rosenthal, Sebastiani, & Stoyanov, 2016; Rosenthal, Farra, & Nakov, 2017). Based on data from social media platforms, the goal is to automatically recognise the sentiment which a piece of writing, e.g. a product review, expresses towards a topic, a product, an event etc. Possible sentiment classes can be, amongst others, simply *positive* versus *negative* versus *neutral* (Go, Bhayani, & Huang, 2009). They can also be more fine-grained: Purver and Battersby (2012) use the six sentiments *happy*, *sad*, *anger*, *fear*, *surprise*, and *disgust*, following theoretical work on emotions (Ekman, 1971). Here also, SVM (Go et al., 2009; Purver & Battersby, 2012) has traditionally been amongst the common choices of classification models. More recently, neural models as well as SVM with neural features have quickly gained popularity (Rosenthal et al., 2017).

Hate Speech Detection Unfortunately, in recent years, social media has also provided a platform for the spread of offensive language and cyber-bullying, and increasing effort has been made to automatically recognise such cases, often referred to as *hate speech* (Schmidt & Wiegand, 2017). Given some writing, the goal is generally to classify it as being offensive or not. The task is not necessarily binary, in many studies (Ruppenhofer, Siegel, & Wiegand, 2018; Gambäck & Sikdar, 2017; Bröckling et al., 2018), authors use more fine-grained class labels indicating the type, target or the severity of the offence. Combinations of neural models or neural features with traditional machine learning algorithms such as SVM and logistic regression have been shown to be effective for this task (Del Vigna, Cimino, Dell'Orletta, Petrocchi, & Tesconi, 2017; Davidson, Warmsley, Macy, & Weber, 2017; Gambäck & Sikdar, 2017).

2.4 Distant Supervision for Social Media Data

While social media platforms give easy access to data, obviously, the data are unannotated, making them not yet suitable for supervised machine learning. Human annotation, however, is always time-consuming and costly. To meet this challenge, several studies have successfully applied *distant supervision* to NLP tasks involving social media data. In distant supervision, (noisily) labelled training data are automatically generated from an existing, unlabelled resource, typically by *automatically labelling samples* based on some heuristics or proxy. To illustrate:

Performing multi-class sentiment classification on Twitter data, Purver and Battersby (2012) examine the use of a) emoticons and b) hashtags as proxies to a tweet's true sentiment label. The emoticons in question would include `:-)`, `:-@`, `:-o` etc., and the relevant hashtags are those which essentially declare emotions, such as *#happy*, *#scared*, or *#sadness*. The authors find that models trained on distantly labelled data, based on either method of distant supervision, do learn to discriminate between sentiments in tweets to a reasonable extent, although performance varies amongst the six emotion classes they use. With respect to happiness, sadness and anger, classifiers trained in this manner perform well, whereas they do less well on the emotions fear, surprise, and disgust. The authors believe that the emoticon and hashtag conventions regarding these classes might be too vague to effectively provide reliable distant supervision.

Emmery, Chrupała, and Daelemans (2017) use distant supervision to predict the gender of Twitter users based on their tweets, which is a part of the author profiling task (see above). To generate gender-labelled training data automatically, they rely on Twitter users' *self-report* of their gender: Using a set of search terms and heuristics, they identify tweets in which users declare their own gender in some reliable way, then pull all Twitter data from these gender-matched users to obtain a gender-labelled dataset. Emmery et al. (2017)'s method of distant supervision is largely adopted for the present thesis. We therefore explain their technique in greater detail in the chapter to come.

Overall, the project described in the present thesis can be placed at the juncture of the above-mentioned fields of automatic prediction of learner levels and NLP on social media data. Using Emmery et al. (2017)'s distant supervision method, we harvest level-annotated data produced by non-native speakers of English, in particular from Twitter and Reddit. Given that these data constitute language production arising from natural communication, produced not as part of some form of task or exercise and without any assessment in mind, we deem such data *spontaneous*. We then apply the task of learner level prediction to these data, also examining whether they can be informed by training signals learned from a more “typical” set of elicited data from the language learning context.

Chapter 3

Data Collection

This chapter provides an overview of our data for the present level prediction project. A central sub-task in this project is the collection of data representing non-native English speakers' spontaneous language production, annotated with their respective level of proficiency. As previously mentioned, for this purpose we harvest social media data from Twitter and Reddit in English which

- we know to be produced by non-native speakers of English and
- of which we have some information concerning the proficiency level of the writer.

Following Emmery et al. (2017)'s approach to distantly supervised gender classification, we first searched Twitter and Reddit and identified a set of users who have made self-reports of their English proficiency level. Subsequently, we extracted all available writings produced by these users and discarded the non-English and otherwise unsuitable data. In contrast to gender, foreign language proficiency is subject to possible changes over time. Therefore, where possible, we also constrained our data collection to a limited period around the time at which the users reported their proficiency level. The following sections describe our data collection process in greater detail.

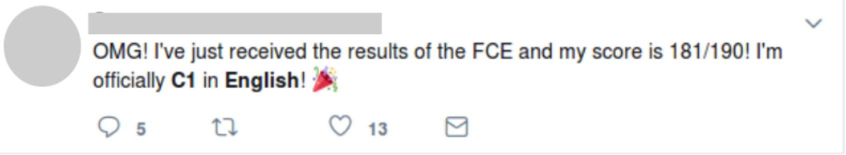
3.1 Twitter

3.1.1 Challenges to Automatic Data Collection

In their task of retrieving tweets containing gender self-reports, Emmery et al. (2017) use a set of simple queries including `{I' / I a}m a {man, woman, male, female, boy, girl, guy, dude, gal`. Moreover, where the query terms appear in the context of a set of linguistic cues such as *as if I'm a girl* or *Don't just assume I'm a guy*, the tweeter's true gender is assumed to be the opposite of what the query term indicates. Finally, the authors remove hits in which the query appears as part of a retweet or a quote to increase the accuracy of their distant gender labels.

In total, Emmery et al. (2017) retrieved 6,610 Twitter users, paired with their self-reported gender. From these users, they extracted a total of 16,788,621 tweets.

Twitter also contains users' self-reports concerning their proficiency levels in English (and other foreign languages), often in status updates in which the tweeters announce their having passed an official exam and achieved a certain level. Figure 3.1 gives an anonymised example.



[redacted]

OMG! I've just received the results of the FCE and my score is 181/190! I'm officially **C1** in **English**! 🎉

FIGURE 3.1: Example proficiency level self-report from Twitter

However, while it would be the ideal approach to largely adopt Emmery et al. (2017)'s automatic method of identifying users of relevance, we quickly realised that it faced the following two difficulties: First, there are larger amounts of ways for users to declare their language levels. While phrases of the form ... *I'm a girl/guy* ..., which are targeted by Emmery et al. (2017)'s queries, can arguably be considered the most generic way of declaring one's gender, there is no such equivalent with regard to reporting one's language proficiency. Common phrases can include *I'm pretty much at B1 in English*, *My English level is C1*, *I speak fluent English (C1)* etc. Therefore, flexible search queries would be required. On the other hand, matching tweets which simply have the words *English*, *I/my* and *A2/B1 etc.* near each other would overproduce and return too many false hits like *I'm a native speaker of English with B2 in German* or *I teach English, from A1 to C1 level.* Thus, in the task, automatic extraction of relevant users would need to be complemented by extensive manual reviewing to ensure that a hit in question indeed involves a proficiency self-report.

Second, and more importantly, Twitter's Standard Search API[1] only supports seven days of history. That is, given a search query, it will limit its search to tweets from the most recent seven days. Preliminary search attempts showed that using the exact string "`I'm B2 in English`" returned no hits at all. A search using the query `I AND B2 AND English`, which simply targets tweets containing these key words, returned 16 hits, of which only four are genuine tweets of interest to the task. The remaining tweets

[1] https://developer.twitter.com/en/docs/tweets/search/overview

represent various types of false hits, including cases in which *B2* does not refer to a CEFR level and cases like *English girl speaking French (B1/B2 level, 95% self-taught)*, where *English* is used as an adjective. Thus, even if searches based on loose, over-generating queries returned an abundance of hits, there would be no straight-forward way to automatically filter out the large portion of false positives. Emmery et al. (2017)'s search was constrained by the same API search limit. However, they were indeed able to gather sufficient users based on the tweets from the most recent seven days. There are simply far fewer cases of users discussing their English language levels than of users making reference to their gender.

For the above reasons, it was revealed that adopting Emmery et al. (2017)'s method of automatically extracting user profiles of interest would not provide sufficient users in our task, due to the nature of our query and the API limit on the search history. Furthermore, extensive manual validation and filtering would be necessary to ensure a high precision of the search results, which is necessary as they function as seeds on the basis of which we would identify relevant users and later extract more data. We therefore decided to conduct the extraction of users and their self-reported proficiency levels manually.

3.1.2 Manual Identification of Proficiency Self-Reports

Instead of using the API, we manually entered a set of queries into the *Search tweets* bar on the Twitter home page. This search method was not limited to only seven days of search history. Our queries are of the following forms: To encourage more hits, we did not search for exact-string matches with phrases, but searched for tweets containing the below key words. The search was case-insensitive.

- `I, English, level, {A1, A2, B1, B2, C1, C2}`
- `I'm, English, {A1, A2, B1, B2, C1, C2}`
- `my, English, {A1, A2, B1, B2, C1 , C2}`

As mentioned, we realise that language proficiency levels can very well change over months and years. Therefore, to prevent using, for instance, self-reported levels from January 2015 as distant labels for tweets from March 2018, we only extracted proficiency self-reports amongst query hits from 1st December 2017 to 18th March 2018. The assumption made is that when we download individual users' most recent tweets at a later stage (in April 2018), the majority of them would reflect the same level of English as that reported by the user between December 2017 and March 2018.

Some self-reports of English proficiency refer to results from some form of formal assessment, be it an established exam or an online test. Examples of such tweets include *Ayyy. I passed my "Language and Use I" exam with a 1.3 (best grade would be 1.0). That means I can now officially brag about my English being C1 level in the CEFR.* Others are statements without reference to anything to back up the assessment, as in *I'm kind of anxious about this account because I don't know how to make friends, I think my english sucks even though I have C1 level and the only thing interesting about myself is my dog.* We make the assumption, however, that users' self-report offer sufficiently reliable proficiency labels.

We indeed retrieved several cases which would have confused automatic searches and would have been difficult to filter out automatically, such as *Wtf did something happen in my sleep? I swear since this morning my english is like A2 level and not B2-C1 ... I must have damaged my brain or something* or *Why do I feel like my English turns into A1 level once again when I am hyped?.* Moreover, as an additional challenge, it appears that *A1* and *A2* can also refer to grades in British schools, as suggested in *Just realised it's be 8 years since I received my O Level results. One of the happiest days of my life, because I got an A1 for English. When never in my life, have I gotten an A for English (except for PSLE lol)* and *Fav subject at school? — English!! Loved it (hence my interest in poetry). Got an A2.*

Where users are undecided and report being between two levels, we consistently recorded their level as the lower one. This was a practical choice in order to obtain a single proficiency label for each user.

Using this manual method, we identified a total of 154 Twitter users with their respective proficiency levels. The set of users is skewed towards the higher proficiency levels, such as C1 and C2, while only two users have been found for level A1. This point will be taken up again in Section 3.3.

Twitter API limits the number of retrievable tweets by a specific user to the 3,200 most recent tweets at the time of extraction[2]. We therefore pulled twice in April 2018 (on the 7th and 30th) and extracted all available tweets from the timeline of the users previouly identified, using unique tweet ids to remove those tweets which appeared in both batches.

In total, we gathered 387,298 tweets. However, the raw dataset contained large amounts of content which would not be relevant to the present project, such as retweets, tweets consisting solely of emoticons or URLs or tweets not primarily written in English. Hence, the following steps were carried out to clean the data:

[2] https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline.html

1. Removal of tweets composed before 1st of September, 2017
2. Removal of tweets starting with *RT*, indicating retweets
3. Removal of user mentions (handles) and URLs from the tweet texts
4. Removal of the hash sign from hashtags in tweets, treating them as simple words
5. Removal of tweets which, after the above steps of cleaning, did not contain at least two adjacent letters. This chiefly targeted tweets which consisted solely of URLs
6. Removal of tweets which have not been identified as being primarily in English by `langdetect`[3] (Shuyo, 2010)

Since, at the previous step of extracting proficiency self-reports, tweets from December 2017 onward were considered, the threshold of September 2017 was chosen to exclude tweets older than three months than the earliest self-reports. In choosing `langdetect`, an easily applicable language identification tool for Python and Java implementations, we again followed Emmery et al. (2017). Apart from filtering out non-English tweets, using `langdetect` also allowed us to discard tweets which, while written in English, were not recognisable as such due to the high abundance of emoticons outweighing the actual words. Examples include the tweet *YOUR OUTFIT!!!!!!*, followed by a string of over 30 heart emojis. Such tweets are largely irrelevant to language proficiency assessment, would likely pose an unnecessary challenge to later processing of the tweet data (e.g. part-of-speech tagging) and could therefore be justifiably removed. After cleaning, the original dataset was reduced to 107,767 tweets in total.

3.2 Reddit

Proficiency self-reports on Reddit were collected automatically, using PRAW, the Python Reddit API Wrapper[4], and manually validated. Reddit users discuss a wide range of topics in topic-specific forums, known as "subreddits". As the starting point of our Reddit data collection process, we identified four subreddits[5] dedicated to foreign language or English learning in which users provide self-reported information on their proficiency in English, often with the aim of finding conversation partners for language practice. The three methods described in the sections to come were used to find user

[3] https://github.com/Mimino666/langdetect

[4] http://praw.readthedocs.io/
https://github.com/praw-dev/praw

[5] https://www.reddit.com/r/languagelearning/
https://www.reddit.com/r/EnglishLearning/
https://www.reddit.com/r/language_exchange/
https://www.reddit.com/r/LanguageBuds/

self-reports of their English proficiency levels in these subreddits. As in the case of data collection from Twitter, we took into account the date associated with each proficiency self-report, which at a later stage allowed us to extract user-specific texts produced within a certain time period around that date.

3.2.1 Keyword Search Through User Flairs

Reddit users in specific subreddits can attach to their names a tag known as "flair", which provides some piece of information of relevance to the topic of the subreddit. In the *languagelearning* subreddit, various users use it to indicate their proficiency level in any language they know, for instance `IT N | EN B2 | FR A2 | DE A2` or `ISL(N) | ENG (C2) | ESP (C1) | TUR (B2) | NAV (A1) | GER (A2)`. We used regular expressions to identify users who provide their English level in their user flairs and where the level is not given as *N* or *Native*. For each user, we recorded the date of the submission in which the user used the relevant flair in his/her name, treating it as the time of the proficiency self-report.

The search matched a total of 28 users. Manual inspection of these search hits revealed that some users have included *English* in their flair text but have either not provided a proficiency level at all or an uninformative one which we were unable to convert to the CEFR system, such as *[?]* or *L2*, which presumably indicates a second-language or Level 2 on an unspecified scale. These hits were discarded.

Some users specified their proficiency levels as *beginnner (beg), intermediate (int)* or *advanced (adv)*, which could be deemed parallel to the three CEFR level groups A (Basic User), B (Independent User) and C (Proficient User) (see Chapter 2). In keeping with our practice on Twitter data of taking the lower level where users report being between two levels, we therefore converted *beginner, intermediate* and *advanced* to A1, B1 and C1, respectively. One user indicated his/her level as *CAE (192/200)*, which translates to C1 according to Cambridge Assessment[6].

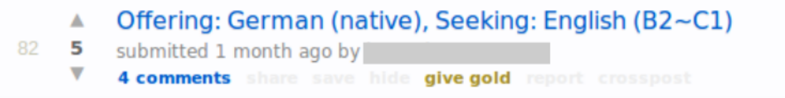
After manual filtering, we obtained 19 users in total using this method.

3.2.2 Keyword Search in Submission Titles

The subreddits *LanguageBuds* and *language_exchange* offer their members the opportunity to find partners for tandem language practice. To facilitate their search, almost all submissions mention in the submission titles the languages they speak and their proficiency level in those languages. Examples include *Offering: Polish(Native), Seeking:*

[6] http://www.cambridgeenglish.org/exams-and-tests/advanced/

English(B1/B2) and *Offering: English (Advanced), Bahasa Indonesia (Native), Seeking: Korea Or Japanese*. Figure 3.2 gives a screenshot example from the *LanguageBuds* subreddit.



82 ▲ 5 ▼

Offering: German (native), Seeking: English (B2~C1)

submitted 1 month ago by [redacted]

4 comments share save hide **give gold** report crosspost

Figure 3.2: Example self-report from Reddit

First, it was thus possible to read-off learners' self-reported English proficiency from the level of English which they offer. Second, it can be assumed that in language practice, learners will seek conversation partners whose skills in the relevant language are on approximately the same level as their own (or slightly superior). Hence, where they indicate a specific level when seeking a language partner for English, as in the above example, that level can be taken as a self-report of their own proficiency in English.

Searching through all available posts from the subreddits *language_exchange* and *LanguageBuds*, we used regular expressions to extract submission titles which contained the word *English* while not preceded or followed by the word *native*, indicating native speakers of English. Once again, we recorded the date of submission for each search hit.

Our search returned 378 hits. Again, they were manually validated and a sizeable portion was filtered out, leaving 216 users with their respective proficiency labels. In the case of most false hits, the user was a native speaker of English but indicated this in a manner which was not picked up by the regular expression. Amongst the users discarded were also those who indicated their level of English as *fluent*, which was considered too vague to be matched with any one of the CEFR levels. As done previously, the levels *beginner*, *intermediate* and *advanced* were mapped to A1, B1 and C1, respectively. Finally, we ensured that there was no overlapping between the users found through this method and those found through user flairs, thus barring the possibility of duplicates amongst our set of users.

3.2.3 General Phrase Search in Submissions

The last of the four subreddits, *EnglishLearning*, is a forum in which learners of English pose in their submissions a variety of questions concerning the language, such as *I have a question about position of preposition in relative clauses*. However, there is largely no necessity for the users to specify their own level of proficiency for this purpose. Hence, despite being most directly concerned with English learning, this subreddit did not lend itself easily to the purpose of automatically identifying users with their self-reported levels. We therefore searched through the titles and textual bodies of all available submissions, using a series of regular expressions to extract occurrences of phrases like *I have level B1 in English* or *My English is at level C1*. In order to increase the chance of finding more user self-reports, the search expressions used were aimed at increasing recall at the expense of precision, which, again, entailed the necessity of extensive manual validation to filter out irrelevant search hits. This filtering as well as conversion of some proficiency self-reports to the CEFR system was carried out in the same manner as previously described.

Apart from the subreddit *EnglishLearning*, this most general method was also applied again to the previous three subreddits as well as a fifth subreddit called *language*, which, however, did not produce any search hits. Once again, for each search hit the time of submission was recorded, and it was ensured that only users who had not already been extracted through any of the previous methods was retained. Overall, this method yielded 122 raw search hits, of which 42 were revealed to be genuine and useful additions to our set of users.

3.2.4 User-Based Data Extraction

Based on the variety of methods described above, a total of 277 unique users with their respective self-reported proficiency levels were found. Using PRAW, we pulled from each user all of their available writings on Reddit, consisting of original submissions and comments in any subreddit they have contributed to. As mentioned, for all users, we recorded the date associated with their self-reported proficiency level. We now used it to exclude writings produced more than three months around the date of self-report to avoid sampling data which could not reliably be associated with the proficiency levels which users had reported of themselves. Apart from this time-based filtering, we performed similar steps of cleaning as in the case of the Twitter dataset, removing URLs and discarding whole samples which consist solely of non-letters (such as emoticons and punctuation marks) and those which are not identified as written in English by `langdetect` (Shuyo, 2010). Samples shorter than 30 characters, which affected 1,752 samples, were also removed.

Furthermore, once again following Emmery et al. (2017)'s methods, we removed from the resulting dataset all of the "self-report samples", i.e. those samples which contained the self-report through which we identified the users and their respective levels. This was done through the unique ID of each self-report sample. We removed these self-report samples as it would likely inflate the classification results if many samples contained a direct statement of the true proficiency labels. The same was not carried out in the case of the Twitter dataset since the users and their corresponding levels were obtained manually instead of through the API (Section 3.1.2) and it was therefore difficult to obtain the IDs of each individual self-report sample.

From Reddit we collected an inital set of 17,075 samples. After the above-mentioned selection and cleaning, we obtained a total of 10,371 level-labelled samples as our final dataset from Reddit.

3.3 Complete Dataset of Spontaneous Language

Our complete set of level-labelled social media dataset consists of 118,138 samples, of which 107,767 were harvested from Twitter and 10,371 from Reddit. Table 3.1 below summarise some aspects of the dataset:

	Twitter	**Reddit**	**Total**
Number of samples	107,767	10,371	118,138
Mean characters per sample	81.09 (SD = 63.47)	221.55 (SD = 370.85)	93.42 (SD = 131.63)
Mean words per sample	17.70 (SD = 14.07)	47.90 (SD = 78.69)	20.35 (SD = 28.23)

Table 3.1: Core characteristics of the social media dataset

Particularly in the case of Reddit data, the samples vary extremely in length despite our removal of extremely short samples. This is unsurprising as writings consist of both comments, which can be no more than one to two sentences, and original submissions, which can be texts of several paragraphs. We considered splitting long samples into multiple samples with the same proficiency label but opted against such a permanent alteration of the dataset out of the following consideration: Were samples to be split at this stage, they would be stored in the dataset as independent samples. If the same original sample were to have a section of it in the eventual training set and another in the test set, the classification result might be biased. Therefore, splitting long samples was carried out at a later point within the training dataset (Section 4.1).

The distribution of the the six proficiency levels in the Twitter, Reddit and the joint social media dataset are as shown in Table 3.2:

	Twitter	**Reddit**	**Total**
A1	908	20	928
A2	1,210	86	1,296
B1	10,550	639	11,189
B2	12,344	2,609	14,953
C1	50,836	4,540	55,376
C2	31,919	2,477	34,396

Table 3.2: Distribution of the six levels in the social media data

Clearly, the dataset is highly skewed towards the higher levels, with the most frequent class being C1 and with significant under-representation of the classes A1 and A2, particularly the former. Reasons for this are obvious: First, non-native speakers are far less likely to either advertise / report their levels on social media or to engage in tandem language learning when their level is quite low, possibly too low for working conversations. Second, those who do report their beginner-level English proficiency do not otherwise write in English, meaning that a large amount of the data that we do extract from their accounts is discarded as non-English data. This imbalance of level distribution is certainly not optimal. Yet, given the self-report-based method of data collection, it is difficult to alleviate. Future improvements in this respect would undoubtedly be welcome.

3.4 Dataset of Elicited Language: Efcamdat

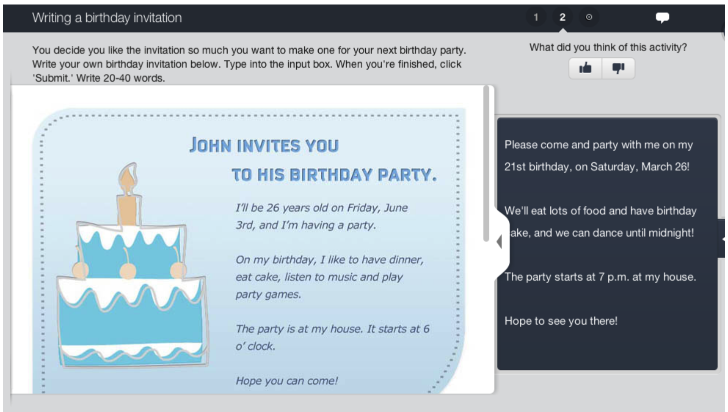
3.4.1 EF-Cambridge Open Language Database

The above sections detail our acquisition of level-labelled *spontaneous* data by non-native speakers, which are to be juxtaposed to *elicited* data produced in the context of foreign language learning. For the side of elicited data, we drew data from a readily available corpus of learner English, the EF-Cambridge Open Language Database (Efcamdat) (Geertzen, Alexopoulou, & Korhonen, 2013; Huang, Murakami, Alexopoulou, & Korhonen, 2018). Efcamdat is a POS-tagged, dependency-parsed and partly error-annotated English learner corpus created at the Department of Theoretical and Applied Linguistics of Cambridge University, in conjunction with Education First (EF). It comprises written data by English learners at a range of different levels. The second and latest distribution, which was used in our studies, is publicly available on the website of Efcamdat[7].

Geertzen et al. (2013) give the following characteristics on the data: The corpus contains over 500,000 written essays (approximately 33 million sentences) produced and submitted by English learners from around the world as part of their studies in EF's online English school *Englishtown*, later renamed *English Live*[8]. Courses of 16 proficiency levels are offered by the online school, with each level consisting of up to eight study units. At the end of each such unit, learners are required to compose an essay or piece of text for which they are given the topic as well as a model answer. Figure 3.3 gives an example of a writing prompt from the Level 2 material.

[7] https://corpus.mml.cam.ac.uk/efcamdat2/public_html/

[8] https://englishlive.ef.com/en-gb/


Writing a birthday invitation
1
2
What did you think of this activity?

You decide you like the invitation so much you want to make one for your next birthday party. Write your own birthday invitation below. Type into the input box. When you're finished, click 'Submit.' Write 20-40 words.

JOHN INVITES YOU
TO HIS BIRTHDAY PARTY.

I'll be 26 years old on Friday, June 3rd, and I'm having a party.

On my birthday, I like to have dinner, eat cake, listen to music and play party games.

The party is at my house. It starts at 6 o' clock.

Hope you can come!

Please come and party with me on my 21st birthday, on Saturday, March 26!

We'll eat lots of food and have birthday ake, and we can dance until midnight!

The party starts at 7 p.m. at my house.

Hope to see you there!

FIGURE 3.3: Example prompt with user interface for a Level 2 task

Evidently, data from this source can be regarded as elicited and controlled: Writers produce their texts in response to a specific prompt and instructions, with a given topic and even an example piece of writing.

According to Geertzen et al. (2013), the 16 proficiency levels used by *Englishtown* are comparable with widely-known measures of proficiency, including the CEFR levels, and can be aligned to them as shown in Table 3.3 (reproduced and adapted from their original paper):

Englishtown	1-3	4-6	7-9	10-12	13-15	16
Cambridge ESOL	-	KET	PET	FCE	CAE	-
IELTS	-	<3	4-5	5-6	6-7	>7
TOEFL iBT	-	-	57-86	87-109	110-120	-
TOEIC Listening & Reading	120-220	225-545	550-780	785-940	945	-
TOEIC Speaking & Writing	40-70	80-110	120-140	150-190	200	-
CEFR	**A1**	**A2**	**B1**	**B2**	**C1**	**C2**

TABLE 3.3: Alignment of *Englishtown*'s 16 levels with standard proficiency measures

Upon enrolment at the online school, students are allocated to one of the levels through a placement test and can then work their way to the higher levels.

Efcamdat has been lemmatised, POS-tagged using the Penn Treebank Tagset (Marcus, Marcinkiewicz, & Santorini, 1993) and syntactically analysed with dependency parsing (De Marneffe & Manning, 2008). Furthermore, it is (partially) an error-annotated learner corpus in which parts of the data (36% in the case of the database's first release) have been manually annotated with the learners' errors and corresponding corrections (Geertzen et al., 2013). Moreover, though not necessarily relevant to the present study, the meta-data available in Efcamdat include unique IDs for all learners, their nationalities and the mark they received. Alexopoulou, Geertzen, Korhonen, and Meurers (2015) remark that where learners' country of residence and nationality match and the country in question has a clearly dominant and/or official language, their nationality can be considered an approximation to their native language. Such information makes the dataset suitable for cross-sectional and longitudinal research on second language acquisition and experiments in grammatical error correction and automatic grading. For comparability of our spontaneous and elicited datasets in the present study, however, we made use only of the plain texts without error-annotation and parse information, along with the course/proficiency level, mapped to the CEFR scale according to Table 3.3.

3.4.2 Our Dataset from Efcamdat

Scripts from the Efcamdat database can be exported as XML-files (Geertzen et al., 2013). Its web-interface allows for selection of scripts based on the level, the study units and the learners' nationality. Our selection of Efcamdat data were drawn from speakers from China, Germany, Japan, Mexico, Russia and Saudi Arabia. For one thing, they count amongst the nationalities with most learners in the database. Moreover, these learners' native languages (L1), approximated by their nationality, cover a range of different language families. Learners with different L1s might make different types of errors (see for instance Chierchia (1998) and DeKeyser (2005) for a discussion on the acquisition of English articles by speakers with different L1s), and we did not wish to limit the proficiency classification task to learners from a specific L1 background when having information on their likely L1 at our disposal. With respect to level and study units, where such an amount of scripts was available, we randomly selected 250 scripts for each study unit at each of the 16 levels[9]. Where there were fewer than 250 scripts for a given unit at a given level, we took all available ones. All scripts were exported as XML-files and parsed with the Python tool *BeautifulSoup* (Richardson, 2013).

In total, our dataset of elicited data, exported from Efcamdat and mapped to the CEFR levels, consisted of 28,029 scripts / samples. Table 3.4 displays some key figures concerning them, juxtaposed to those of our full social media dataset in italics, repeated from Table 3.1:

	Elicited (Efcamdat)	***Spontaneous***
Number of samples	28,029	*118,138*
Mean characters per sample	559.62 (SD = 324.40)	*93.42 (SD = 131.63)*
Mean words per sample	114.59 (SD = 62.38)	*20.35 (SD = 28.23)*

Table 3.4: Core characteristics of our elicited dataset from Efcamdat, compared to those of the spontaneous dataset from social media

Evidently, samples from Efcamdat are, on average, significantly longer than samples from social media, with learners at the higher levels generally writing more (Geertzen et

[9] Recall that each level consists of eight study units

al., 2013). Similarly as in the case of particularly long samples in the spontaneous data, we considered splitting them. This would also be fruitful since long samples were more likely found at the higher proficiency levels, where we also had fewer samples available (see below). However, due to the same aforesaid reasons, we chose to do so only at a later stage to ensure that a sample would not have parts of it in the training and parts in the testing data.

Overall, the distribution of the six CEFR levels in our Efcamdat dataset are as shown in Table 3.5 (once again contrasted to those in the social media dataset in italics):

	Elicited (Efcamdat)	***Spontaneous***
A1	6,000	*928*
A2	6,000	*1,296*
B1	6,000	*11,189*
B2	5,650	*14,953*
C1	3,749	*55,376*
C2	630	*34,396*

Table 3.5: Distribution of the six levels in the Efcamdat data, juxtaposed to those in the social media dataset

Unlike in the case of the spontaneous data, our elicited dataset is skewed towards the lower levels for the following reason: Overall, only few learners completed their course with *EnglishTown* and reach the highest Level, 16 (Geertzen et al., 2013), meaning that there is generally much more data available at the lower than the higher end. Furthermore, as shown in Table 3.3, in contrast to the CEFR levels A1 - C1, C2 corresponds to only one instead of three *EnglishTown* levels, resulting in it being a clear minority class. Thus, unfortunately though inevitably, our spontaneous and our elicited datasets are skewed in opposite directions.

3.4.3 Topic Influence in Efcamdat Dataset

One particular trait of Efcamdat data merits special attention in the context of this study: As a rule, each study unit, for which a learner writes one script (unless failing and retaking the unit), addresses one specific topic as its essay topic. They range from presenting oneself and one's family at the lowest proficiency level up to discussing news stories at the higher ones. Geertzen et al. (2013) provide some example essay topics at different levels, reproduced and adapted in Table 3.6.

The thorny issue here is that, aside from differing in the proficiency levels which the respective writers display, samples from different *EnglishTown* and hence CEFR levels will also significantly differ in the topics they deal with. Such topical differences are easily captured by content words. Should word n-gram features be used in a proficiency level

Level	Topic	Level	Topic
1	Introducing yourself by email	7	Giving instructions to play a game
1	Writing an online profile	8	Reviewing a song for a website
2	Describing your favourite day	9	Writing an apology email
2	Telling someone what you're doing	11	Writing a movie review
2	Describing your family's eating habits	12	Turning down an invitation
3	Replying to a new penpal	13	Giving advice about budgeting
4	Writing about what you do	15	Covering a news story
6	Writing a resume	16	Researching a legendary creature

Table 3.6: Example essay topics at a range of *EnglishTown* levels

classifier, the system could easily be misled to model the *topical* instead of the *linguistic* distinctions at the different levels. This is likely characteristic of data acquired in a language learning context, where writings are often produced in an elicited manner in response to a specific topic.

At the stage of data collection, we saw no way of counter-balancing this topic influence other than making sure that our data cover as many possible topics at each level as there are in order not to make any particular topic strikingly indicative of a specific level. However, this topic influence will be addressed again in the next chapter, where we discuss methods to mitigate it.

3.5 Chapter Conclusion

We described in this chapter our data collection process. Using users' self-reported proficiency levels as distant labels, we obtained a set of 118,138 samples from Twitter and Reddit, representing spontaneous writings by non-native speakers of English. The process was partly manual and partly automatic with manual validation. On the side of elicited data from a language learning context, we extracted a 28,029-sample dataset from the openly accessible learner corpus Efcamdat, which consists of scripts by learners of English written in the context of an online English course. In the chapters to come, we proceed to detailing our level prediction experiments using these data.

Chapter 4

Level Classification Using SVM and Logistic Regression

This chapter describes and discusses our main experiments of performing level classification in both the elicited Efcamdat dataset and the spontaneous dataset gathered from social media, with a greater focus on the latter. We also discuss whether classification *across* these two domains is possible and look at classification in the *mixed*-domain dataset formed by the union between the two sets. Finally, we also explore a customised evaluation metric that reflects the ordered nature of our six labels. To repeat, the classification task is carried out on a per-sample basis. That is, given a piece of writing, a learned model assigns to it one of the six CEFR proficiency levels.

4.1 Classification on Efcamdat Data

4.1.1 Methods

Following previous literature in which SVM (Cortes & Vapnik, 1995) has been shown to be effective in classification tasks such as author profiling (Basile et al., 2017) and offensive speech detection (Del Vigna et al., 2017; Davidson et al., 2017), we chose to conduct our level classification task using a linear SVM with suface-level linguistic features.

4.1.1.1 Pre-processing

As a simple pre-processing step, we removed from the samples control characters indicating new line, tabs etc. by taking out those characters whose unicode category begins with `C`. Control characters fall into this group[1].

Furthermore, as mentioned in Chapter 3, we planned to split particularly long samples into two to reduce the dataset's variance in sample length. This had the following additional advantage: Long samples are generally members of the higher proficiency classes. Given that the Efcamdat dataset is skewed towards the lower proficiency levels (Section 3.4), splitting the long samples increased the number of samples in the under-represented, higher proficiency levels. In concrete terms, given that the mean sample length of the dataset is 559.62 characters ($SD = 324.4$) (see Section 3.4), we considered samples with more than 800 characters in length to be long samples and split them, giving both resulting samples the class label which the original long sample had. While doing so, sentence boundaries were respected. We used NLTK's (Bird & Loper, 2004) sentence tokeniser to detect sentence boundaries and made sure that splitting samples did not "cut through" sentences. Sample splitting was only performed on the training portion of the dataset, while long samples in the test portion were left unchanged [2]. We opted for this because we ultimately wished to obtain a single proficiency label for a given test sample. Should the long test sample have been split and its two parts receive different class predictions, we would need to address how to decide on a single final label, which we deemed a complication not worth adding.

4.1.1.2 Features

We used a simple set of surface-level linguistic features consisting of the following:

- Unweighted word uni grams
- Unweighted character n-grams in the range between 3 and 6
- Average sentence length in terms of number of characters. For this feature we again used NLTK's sentence tokeniser, obtained the length of each sentence in the sample and took the mean.

Conceptually, word unigrams can reflect the language user's vocabulary size and lexical diversity, which in turn provides information on the person's proficiency level (Crossley

[1] `http://www.unicode.org/reports/tr44/#General_Category_Values`

[2] See below for the train/test division

et al., 2012; Yannakoudakis et al., 2011): Intuitively, rare, formal and domain-specific words are more likely to be used by more advanced users, while beginner-level users likely have a limited vocabulary at their command which consists of common-place and informal terms. The character n-grams were expected to target orthographic errors, which can be characteristic of lower-level non-native language users (Hancke & Meurers, 2013). Finally, sentence length has been shown to be an indicator of language learner proficiency levels (Tack et al., 2017), being in many cases also a proxy to syntactic complexity. We did not discard stop words since they typically include function words such as articles, auxiliary verbs and prepositions (see, for instance, those provided by the Python module `stop-words`[3]), and research in second language acquisition in English have shown errors related to them as characteristic of learner writings (Leacock et al., 2010; Master, 1997).

As discussed extensively in Chapter 3, an undesirable characteristic of the Efcamdat dataset is that the systems are easily misled to model the *topical* differences between the six levels as *English Town*/*English Live* students at different levels are given different topics to write about. While we counteracted this effect at the data collection stage by drawing data from as many topics as were available in the corpus for each level, there is little doubt that this topic influence remained present. We therefore tried to mitigate this by replacing the words in a sample with their part-of-speech (POS) tags, thereby removing actual lexical content. This is related to the techniques used by Goot, Ljubešić, Matroos, Nissim, and Plank (2018), which they call the *bleaching* of text. We adopt this term and refer to our transformations *POS-bleaching* hereafter.

To obtain reliable POS tags, we applied NLTK's (Bird & Loper, 2004) off-the-shelf tagger using the widely-used Penn Treebank tagset[4] (Marcus et al., 1993). We then experimented with POS-bleaching of our samples in the following conditions:

1. Applying POS-bleaching to *all* tokens of the sample
2. Applying POS-bleaching only to nouns and verbs, hence those words whose POS-tag starts with `NN` or `VB`
3. Applying POS-bleaching only to nouns

Nouns and verbs were chosen for POS-bleaching since we deemed them most likely to exhibit topic influence. One disadvantage of bleaching verbs, however, is that the tagger does not distinguish between content verbs and auxiliaries. The latter, however, are not

[3] `https://pypi.org/project/stop-words/`

[4] `https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`

only fully topic-independent, but might provide useful training signals as their irregular inflection systems are a possible source of errors for learners.

An additional piece of information which POS-bleaching unfortunately discards is the general frequency of the words used in a sample. As mentioned above, less frequent words can be indicative of a writer's larger vocabulary and thereby higher level of proficiency. To compensate for this loss, in combination with POS-bleaching we introduced an additional feature for explicit encoding of a word's general frequency. This was done using Speer, Chin, Lin, Jewett, and Nathan (2017)'s tool `wordfreq`[5], which allows lookups of frequencies of words in 36 languages, based on a wide range of data, including Wikipedia, Google Books, News, Twitter etc. They also provide a function which returns a word's *Zipf*-frequency, with values between 0 and 8, where higher values correspond to higher frequencies. For instance, the values 7.75 and 3.75 are given as the respective Zipf frequencies of the words *the* and *monarchy*.

For each word to which POS-bleaching was applied, we looked up their Zipf frequency value and rounded it to a whole number, which effectively meant that there were eight available frequency classes, ranging from 0 for words unknown to `wordfreq` to 8 for highly frequent words like *the*. We then attached this rounded value to the POS-tag of the bleached word. To illustrate, the nouns *cats* and *man* would be transformed to the strings *NNS_4* and *NN_6*, respectively. Finally, we took word n-grams in the range between 1 and 3 on the POS-bleached and transformed sample texts. Character n-grams were not used when bleaching was applied.

4.1.1.3 Model and Set-up

We used a linear SVM, implemented with Python and the Python machine learning toolkit *Scikit-learn* (Pedregosa et al., 2011). Specifically, we used their `LinearSVC` model[6], a more efficient implementation of linear SVMs. The package's default hyperparameters were adopted, which include L2 regularisation and `C = 1.0`. Given that SVM is inherently a binary classifier, the model takes the one-versus-rest strategy (as opposed to the computationally more expensive one-versus-one strategy) as its default for performing multi-class classification (which applies in the present task).

We randomly carved out 25% of the dataset to be our test set and trained on the remaining 75%. This same procedure was adopted for all experiments reported on in this chapter.

[5] https://github.com/LuminosoInsight/wordfreq

[6] http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

4.1.2 Results and Discussion

Given the clean nature of the dataset and the topic influence, we expect classification on the Efcamdat dataset to achieve good results. An overview of these figures are given in this section, using different combinations of pre-processing steps and features. We provide the results (Table 4.1) in terms of accuracy and macro-average F1 measures, given in percentages. We also set them in contrast to the performance of a baseline system based on the majority class in the training split of the dataset. This majority class is generally `A1`, `A2` or `B1`, given that these three classes are equal in size and jointly the most frequent class in our *full* Efcamdat dataset (see section 3.4).

System + features	**Acc**	**Macro F1**
SVM + splitting long samples + word + char n-grams	91.9	90.7
SVM + *no* splitting samples + word + char n-grams	**92.3**	**90.8**
SVM + word + char n-grams + mean sentence length	92.1	90.6
SVM + POS-bleaching of *all* words	56.5	46.1
SVM + POS-bleaching of *nouns* + *verbs*	82.7	77.5
SVM + POS-bleaching of *nouns*	86.3	81.9
SVM + POS-bleaching of nouns + freq. of bleached words	87.6	83.2
Majority class baseline	20.8	5.7

Table 4.1: Overview of classification results on the Efcamdat dataset based on random 75%/25%-train/test splits, with best results highlighted; figures given in percentages

As expected, the results on the Efcamdat data are extremely good, with both accuracy and F1 scores over 90% in the best condition, which used little pre-processing and only word and character n-grams (highlighted). Table 4.2 provides an example confusion matrix based on the performance of the highest-scoring SVM (highlighted in Table 4.1). The vertical labels represent the true gold-standard class labels and the horizontal labels the system-predicted labels.

	A1	A2	B1	B2	C1	C2
A1	**1,475**	17	9	5	1	0
A2	40	**1,390**	41	21	9	0
B1	20	42	**1,387**	37	18	0
B2	4	24	62	**1,266**	25	2
C1	4	16	53	48	**833**	2
C2	4	3	6	17	11	**116**

Table 4.2: Example confusion matrix for classification on EFcamdat using an SVM with word and character n-grams

Overall, the confusion matrix matches the high scores reported in Table 4.1. No noteworthy patterns are discernible among the small number of errors made by the system.

Contrary to what we predicted, splitting long samples into two in the training set did not seem to have any positive effect; we therefore removed this pre-processing step in all other runs on this dataset. Similarly, the average sentence length of a sample also failed to produce any noticeable gains. It should be noted, however, that when the system was run with average sentence length as the *only* feature, the result was significantly above the majority class baseline (acc: 30.8%, macro F1: 14.7%), confirming that it is indeed a useful feature. Possibly, it is simply not significant enough to raise the performance based on the high-dimensional sparse n-gram features, which is already extremely high.

Certainly, the scores achieved by the best-performing model here are inflated due to the topic influence characterising the Efcamdat dataset. POS-bleaching was added to mitigate this effect. We see that bleaching out all words, as can be expected, dramatically worsens the performance, although it is still very much above the baseline. When POS-bleaching is applied to verbs and nouns only, or indeed only to nouns (arguably the most topic-specific POS-category), accuracy and macro F1 scores are again over 80%. It is also shown that the addition of Zipf frequency information to the bleached nouns is helpful. However, whether these results suggest that POS-bleaching indeed effectively combats topic influence and possibly makes a model less dependent on the training dataset will be taken up again in due course.

4.2 Classification on Social Media Data

4.2.1 Methods

For the classification task on the social media dataset (which can be expected to be a harder problem due to the noisy nature of both the data and the labels), we proceeded along the same lines as in the case of the Efcamdat data but experimented with some additions. We again used a linear SVM with a variety of features, not all of which proved to be useful. Moreover, we also experimented with a logistic regression model and in fact found it to be superior. Details of our studies on the Twitter and Reddit data are presented in this section.

4.2.1.1 Pre-processing

We performed similar steps of pre-processing as described in Section 4.1, removing control characters and levelling the dataset's high variance in sample length by splitting long samples into two within the training portion. Samples longer than 400 characters in length were chosen for splitting. Due to Twitter's character limit[7], this chiefly affected samples from the Reddit section of the social media dataset, which varies particularly highly in length (Section 3.3).

[7] https://developer.twitter.com/en/docs/basics/counting-characters.html

In addition to the above, we used regular expressions to remove emoticons: We matched characters in the unicode range `\U00010000-\U0010ffff`[8] as well as strings which matched the Regex `:-?[\)\(9D/P3]|[oO]\.[oO]`. These two expressions recovered most (albeit not all) emoticons in the samples. The emoticons matched were not discarded, but replaced with the generic symbol "e", indicating the presence of an emoticon. We were motivated by the consideration that, while emoticons do not bear any direct relation to people's proficiency in a given language, they do reflect their handling of communication on social media, and we deemed it possible that they might contain useful signals for our task.

Moreover, under the assumption that social media users might intentionally use conventionalised web language acronyms and abbreviations which ought not to be mistaken as orthographic errors, we normalised them by "spelling them out". For this, we used a list of web language abbreviations found on the website *socialreport*[9] and adopted the 45 acronyms listed under the category "Fun Acronyms For Daily Use". They include, for instance, *BFF* for *Best Friends Forever* or *JK* for *Just Kidding.*

4.2.1.2 Features and Various Experimental Settings

Surface-Level Features The main surface-level features used were again identical to those used in classifying the Efcamdat dataset, viz. word unigrams, characters n-grams in the range of 3 to 6, and the mean sentence length in a sample. Furthermore, we experimented with using word bigrams in addition (which enlarged the feature space and slowed down training).

Dimensionality Reduction Due to the noisy and open-domain nature of our social media dataset and the resulting large vocabulary, we expected to obtain a large and sparse feature space based on the word and character n-gram features. Therefore, we explored the effect of applying dimensionality reduction to the feature space prior to classification. Since the Scikit-learn toolkit does not support the application of the widely-used Principal Component Analysis (PCA) (Jolliffe & Cadima, 2016) to sparse feature matrices, we chose to use Latent Semantic Analysis (LSA) via Single Value Decomposition (SVD) (Schütze, Manning, & Raghavan, 2008), with the number of output dimensions d set to $d = \{100, 1000\}$. Scikit-Learn's default parameters[10] were used in the implementation.

[8] https://unicode.org/emoji/charts/full-emoji-list.html

[9] https://www.socialreport.com/insights/article/115003187266-80-Social-Media-Acronyms-You-Need-To-Know

[10] http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html

Over- and Undersampling To recall, the social media dataset is highly skewed towards the higher proficiency levels, owing to the manner in which it was collected. On the extreme end, 56,109 samples from the largest class `C1` are contrasted with only 935 samples from the smallest class `A1` (Section 3.3). Given this class imbalance, we experimented with performing over- and undersampling on the training data using *Imbalanced-Learn* (Lemaître, Nogueira, & Aridas, 2017), an extension to Scikit-learn designed to deal with imbalanced data. Specifically, we experimented with

1. *Oversampling* to the largest class (viz. `C1`) using random oversampling[11]: In all but the majority class (viz. all but `C1`), random copies of the original samples are made for the class to reach the size of the larget class. The total number of samples after oversampling is the size of the majority class multiplied by the number of classes, hence 336,654 in our case.
2. *Undersampling* to the smallest class (viz. `A1`) using random undersampling[12]: Assume that the smallest class has N samples in total, the under-sampler reduces the size of all other classes to N samples by randomly picking N samples from each. The total number of samples is the size of the smallest class multiplied by the number of classes, hence 5,610 in our case.

Notice that over- and under-sampling was applied only to the training portion of the data. The highly skewed class distribution remained in the test data portion.

Fewer Classes Given that six proficiency levels constitute a comparatively fine level distinction, we also investigated classification using *fewer* than the six levels. In particular:

1. We ran a system based on a coarser level distinction consisting of three classes. For this, the levels A1 and A2, B1 and B2, and C1 and C2 were conflated to the levels A, B, and C. Recall that at the data collection stage (Chapter 3), some of the Twitter and Reddit users in fact reported on their levels in terms of the three-part proficiency levels *beginner* - *intermediate* - *advanced* to start with, which we had converted to A1, B1 and C1.
2. Furthermore, on the assumption that adjacent classes on the proficiency scale would be difficult for the classifier to distinguish, we carved out the datasets with

[11] `http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.over_sampling.RandomOverSampler.html`

[12] `http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.under_sampling.RandomUnderSampler.html`

the labels `C1` and `C2` and tested a *binary* classifier on them to see if it would learn the likely limited distinctions between two similar classes. Levels C1 and C2 were chosen since they were the two adjacent levels with the largest amounts of samples available.

4.2.1.3 Models and Set-up

Once again we used Scikit-learn's implementation of the linear SVM with their default parameters. In addition, we also experimented with a logistic regression model instead of the SVM. For the latter, we also employed Scikit-learn's implementation[13]. The default hyper-parameters, which were again kept in place, include L2 regularisation, `C = 1.0` and the use of the one-versus-rest strategy for non-binary classification.

As done previously, all system were tested using a random train/test-split of 75%/25%.

4.2.2 Results and Discussion

An overview of our classification results under the various experimental settings is provided in the tables in this section. We again provide accuracy scores and macro-average F1 measures as percentages. Again, a majority class baseline is provided where all test samples are classified as `C1`.

System + features	**Acc**	**Macro F1**
SVM + splitting long samples + word + char n-grams	56.7	43.2
SVM + *no* splitting samples + word + char n-grams	56.7	43.7
SVM + word uni + char n-grams + mean sentence length	56.8	43.7
SVM + word uni + *bi*grams + char n-grams + sentence length	57.2	43.7
SVM + word uni + char n-grams + LSA ($d = 100$)	48.3	15.8
SVM + word uni + char n-grams + LSA ($d = 1,000$)	53.0	30.0
SVM + word uni + char n-grams + *Over*sampling	56.2	43.4
SVM + word uni + char n-grams + *Under*sampling	27.3	22.6
Majority class baseline	47.2	10.7

Table 4.3: Overview of the SVM's classification results on the social media dataset based on random 75%/25%-train/test splits; figures given in percentages

Table 4.3 shows all the results obtained from the SVM in classifying into the usual six classes. Clearly, the task is significantly harder on the social media dataset. This is as expected, given the noise in the writing, the distantly supervised nature of the labels and the brevity of many samples. However, we see that all systems (with the exception of the undersampled one) do significantly beat the majority class baseline, which is higher

[13] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

than in the Efcamdat dataset as class C1 is the single, clearly most numerous class in this dataset.

The variations in the surface-level features appear to have little effect on the system performance: As in the case of the elicited data, treating particularly long samples in the training set as two separate samples did not produce any gains. Equally making no noteworthy difference is the information on the average sentence length in a sample. Unlike in the case of Efcamdat data, when the SVM is run with average sentence length as the only feature, its performance is equal to the baseline, suggesting that the feature simply does not encode useful information. This could partly be due to the noisier nature of social media data, on which NLTK's sentence tokeniser possibly underperforms. Finally, somewhat to our surprise, the addition of word bigrams also has almost no effect on the classification scores.

What is undoubtedly shown in Table 4.3 is that LSA dimensionality reduction as well as over- and undersampling do not appear to be viable options for system improvement: Dimensionality reduction with either numbers of target dimension (100 and 1,000) affected classification performance negatively; Oversampling to the majority class yielded approximately the same scores as without it, likely because the oversampled classes were simply filled up with copies of existing samples, which did not in fact provide any additional training signals; Undersampling to the smallest class, on the other hand, impacted the performance detrimentally, likely because the resulting dataset after performing undersampling was simply too small.

Once again we provide an example confusion matrix showing the error tendencies for classifications on the social media dataset (Table 4.4). The matrix is based on the top-most model of Table 4.3, viz. an SVM with word and character n-grams.

	A1	A2	B1	B2	C1	C2
A1	**68**	1	11	11	78	52
A2	2	**33**	24	35	149	64
B1	3	21	**1,223**	258	889	431
B2	5	22	235	**1,469**	1,357	689
C1	26	55	708	1,082	**9,256**	2,694
C2	17	44	390	624	2,810	**4,699**

Table 4.4: Example confusion matrix for classification on social media data using an SVM with word and character n-grams

Two remarks are worth mentioning: First, we see the influence of the extreme imbalance of class distribution. Errors are extremely frequent among the heavily under-represented lower levels A1 and A2. More specifically, the majority of falsely classified samples from these classes are mispredicted as class `C1`, which of course is the majority class by far in the social media dataset. Second, among the classes on which classification is more successful, e.g. C1 and C2, most errors are misclassified into the level just above or just below the true level. This reflects the ranked nature of our six classes, which in fact form a proficiency scale. This point will be addressed in greater detail below (Section 4.5).

Classes	**Acc**	**Macro F1**
SVM + word uni + char n-grams		
A, B, C, (using all data)	75.6	52.6
C1, C2, (using C1 and C2 samples)	70.0	68.0
Majority baseline		
A, B, C, (using all data)	75.7	28.7
C1, C2, (using C1 and C2 samples)	62.1	38.3

Table 4.5: Overview of results where a) classification is performed on three classes only and b) classification exclusively aims to distinguish between the classes C1 and C2; contrasted with majority class baseline in the same setting

Table 4.5 shows the results of our experiments using labels other than the six CEFR levels. As could be expected, the coarser classification problem into three levels is an easier task. Due to the heavy over-representation of the class C in this setting (with C1 and C2 being the two largest classes in our collected dataset), the majority class baseline is extremely high in terms of accuracy and is in fact equal to the SVM output. However, the SVM beats it by a wide margin in terms of the macro-average F1 score. We expected the likely limited difference between the levels C1 and C2 to be difficult to learn. Yet, the classifier did manage this to a respectable extent, outperforming the baseline by a large margin.

Finally, our results of classification to the original six proficiency levels using an *alternative* model, the logistic regression model (LogReg), are shown in Table 4.6.

System + features	**Acc**	**Macro F1**
LogReg + word uni + char n-grams	61.7	47.4
LogReg + word uni + bigrams + char n-grams	**63.4**	**47.5**

Table 4.6: Overview of the logistic regressor's classification results on a random train/test-split of 75%25% set, best performing system on the dataset highlighted.

Although SVMs appear to enjoy greater popularity in the literature (Basile et al., 2017; Del Vigna et al., 2017), we find that in our task the logistic regression model in fact significantly outperforms the former with more than four points of improvement in terms of macro-average F1. The beneficial effects of adding bi-grams remains ambiguous, being present in the accuracy scores but absent in the macro F1. Overall, in the social media domain, we obtained the best classification results from the logistic regression model with simple surface-level word and character n-gram features, which are highlighted in Table 4.6.

4.3 Cross-Domain and Mixed-Domain Classification

One of our main interests in the present project is to examine to what extent the elicited data from the controlled language learning environment can inform the noisier social media domain. Therefore, we also conducted various classification experiments *across* the two domains, with an emphasis on improving classification in the social media domain using training signals from the language learning context. Furthermore, we also examined mixing all of our data into a *mixed*-domain dataset and modelling level prediction on that.

4.3.1 Methods

The same methods as in the previous, within-domain tasks were used, excluding those which clearly proved not to be useful, such as dimensionality reduction and over- and downsampling. That is:

Pre-processing We removed control characters from all datasets. In the social media data we additionally normalised the emoticons and web language abbreviations in the manner previously described. The splitting of long samples was abandoned as it accounted for no improvement in either domain.

Features We generally used word unigrams and character n-grams in the range of 3 to 6 and optionally added word bigrams and the mean sentence length feature. We also once again examined the POS-bleaching of nouns, where all words detected as nouns were replaced by their POS-tags and the information on their Zipf frequency (see above). Our original motivation for introducing this feature to the Efcamdat dataset was to make the Efcamdat samples less topic-sensitive by taking out the lexical content of what we considered the most topic-specific category of words, viz. nouns. By reducing the influence of topics, we hoped that models trained on the bleached Efcamdat data would be more transferable to domains where the same co-occurrence between certain topics and certain proficiency levels do not hold. Therefore, testing POS-bleaching in this cross-domain context was of particular interest. Clearly, the POS-bleaching of nouns would need to be applied to both the training domain and the test domain, using the same tagset. Instead of NLTK's POS tagger, in the cross-domain setting we chose to use the POS tagger from SpaCy (Honnibal & Montani, 2017). It is based on a model which is tuned to, among others, web language[14] and could therefore be expected to

[14] https://spacy.io/models/en#en_core_web_sm

perform better on the social media side of the data. The SpaCy tagger also uses the Penn Treebank tagset[15].

Models and baselines The same two models, the linear SVM and the logistic regressor, were once again used. For the *mixed*-domain experiments, once again a single baseline based on the majority class in the training set was used. In the *cross*-domain context, however, we used the following three different baselines:

- Baseline 1: Identical to the baselines previously used, this chooses the majority class in the training data. However, in the cross-domain setting this is bound to be an extremely poor baseline due to the starkly different class distributions in the training and the test domains (Chapter 3). The majority class in the training set, i.e. the full Efcamdat dataset, will be `A1, A2 or B1` (which are identical in frequency and jointly the most frequent class), which are least frequent in the social media test domain.
- Baseline 2: Given that we know class `C1` to be the clear majority class in the *test* domain, i.e. the social media set, this baseline predicts class `C1` for all test samples.
- Baseline 3: The last baseline predicts a class at random.

Training and test data In the cross-domain experiments the main focus was put on training on the full Efcamdat dataset and testing on 5,000 randomly chosen samples from the social media data. However, we also investigated other cross-domain constellations: Amongst others, we held out 5,000 social media samples to test on and *added the remaining social media data to the Efcamdat data to train on.* The model was thus trained on a large, *mixed*-domain dataset and then tested on social media samples. Moreover, given that our social media dataset originates from two different sources, viz. Twitter on the one hand and Reddit on the other, and that they do display different characteristics regarding, for instance, mean and variance of sample length (Section 3.3), we also tested a) to what extent transfer of training signals *between* the two social media subsets of data is possible and b) if either social media dataset gains more from training signals from Efcamdat than the other. For a) we trained on the Twitter and tested on the Reddit domain, and vice versa; for b), we trained on Efcamdat and tested separately on Twitter and Reddit. In each of these cases, the test set consists of 5,000 randomly chosen samples from the relevant test domain.

In the mixed-domain setting, we joined all of our data (from either domain) and again took a random portion of 25% to test on, training on the other 75%.

[15] https://spacy.io/api/annotation

4.3.2 Results and Discussion

Table 4.7 gives an overview of the performance of our models in the cross-domain setting, where we trained on all samples from the Efcamdat domain and tested on a 5,000-sample subset of the social media data. They are contrasted with the scores from the three baselines described in the previous section.

System + features	Acc	Macro F1
SVM + word uni + char n-grams	11.8	9.3
SVM + word uni + bigrams + char n-grams	7.7	7.3
SVM + word uni + bigrams + char n-grams + sentence length	7.0	6.7
LogReg + word uni + char n-grams	**12.1**	**9.5**
LogReg + word uni + bigrams + char n-grams	6.9	6.6
LogReg + word uni + bigrams + char n-grams + sentence length	7.0	6.7
LogReg + word uni + char n-grams + sentence length + POS-bleaching of nouns + freq. of bleached words	5.7	4.9
Baseline 1 (majority class in train)	0.6	0.2
Baseline 2 (always predicting C1)	**47.4**	10.7
Baseline 3 (random prediction)	17.0	**12.8**

Table 4.7: Overview of cross-domain prediction with systems trained on all Efcamdat data and 5,000 randomly chosen social media samples, best non-baseline system and highest overall scores highlighted

It is immediately evident that direct transfer from the Efcamdat to the social media domain is impossible, as shown by the very poor performance of both the SVM and the logistic regressor. In fact, much better performance is achieved by the baselines: The highest accuracy score is obtained by Baseline 2 by always predicting the class known to be the majority class in the test domain, and the highest macro-average F1 score by random prediction (Baseline 3) (Baseline 1 is as poor as expected). In Table 4.8 a confusion matrix based on the best non-baseline system, viz. the logistic regressor with word unigrams and character n-grams, is shown.

	A1	A2	B1	B2	C1	C2
A1	**20**	4	2	6	7	1
A2	26	**9**	4	3	8	1
B1	233	67	**49**	71	59	5
B2	284	97	83	**112**	77	11
C1	942	300	246	429	**376**	48
C2	487	202	153	279	262	**37**

Table 4.8: Example confusion matrix for cross-domain classification using a logistic regressor with word and character n-grams, trained on Efcamdat, tested on social media

Clearly, the most striking observation from the confusion matrix is that social media samples from all proficiency levels are deemed as lower-level writings by a model trained on Efcamdat data. In fact, a large portion of test samples from all classes is predicted as the lowest level A1.

In both machine learning models, the addition of word bigrams and the sentence length feature have a negative effect. A closer examination of the mean sentence length feature reveals the shortcomings of this feature: As a follow-up, we again ran the NLTK's sentence tokeniser and obtained for both the Efcamdat and the social media data the mean sentence length in the samples at each of the six proficiency levels, always in terms of number of characters. The findings are given in Table 4.9.

	A1	**A2**	**B1**	**B2**	**C1**	**C2**
Efcamdat	47.8	57.8	68.3	84.0	89.8	99.9
Social Media	48.4	51.3	51.3	57.6	58.3	57.2

Table 4.9: Mean sentence length in writings at each level for both domains, given in terms of number of characters

Evidently, while in the Efcamdat dataset, rising mean sentence length nicely corresponds to increasing proficiency levels, the same trend is not at all present in the social media dataset, where the figures are remarkably similar across all six levels. This clearly explains why, as previously pointed out, while mean sentence length is a predictive feature in the Efcamdat dataset which gave above-baseline performance when used as the only feature (Section 4.1.2), it was proved entirely unhelpful in the social media domain (Section 4.2.2). Moreover, it is plausible that the addition of this feature in the cross-domain context would have further aggravated the error trend observed in the above confusion matrix: Based on mean sentence length, even social media samples labelled as `C2` would resemble one at level A2 to a model trained on Efcamdat data.

The results of our cross-domain classification tasks in other data constellations are given in Table 4.10. All of these tasks were performed using what has been revealed as the best system. viz. the logistic regression model using only word unigrams and character n-grams in the range of 3 to 6, with the pre-processing steps described in the previous section. We indicate in the table the training and the test set used in each run, along with the accuracy and macro-average F1 scores achieved.

Train	**Test**	**Acc**	**Macro F1**
All data except the 5,000 samples held out for test	Social media (5,000)	63.4	49.0
Twitter	Reddit (5,000)	36.3	16.3
Reddit	Twitter (5,000)	35.0	14.7
Efcamdat	Twitter (5,000)	7.2	6.8
Efcamdat	Reddit (5,000)	12.3	9.4

Table 4.10: Overview of classification results using the same system with different constellations of training and test set

The results shown in Table 4.10 draw attention to two main points: First, when training on a *mixed* set between large parts of the social media data and all of the Efcamdat data and testing on a set of *social media* data only, the performance is hardly better than the best result achieved by training and testing on only the social media data, which, to repeat from Section 4.2.2, reached the accuracy and the macro-average F1 score of 63.4 and 47.5, respectively. Importantly, the model trained on the mixed set has access to significantly *more training data* than a model trained on the social media set only. Namely it trains on a large training portion of the social media dataset, much like the in-domain model, and on all of the Efcamdat data *in addition*. The observation that having 28,029 samples[16] of extra training data only yielded marginal improvement again shows that the Efcamdat domain does not effectively inform the social media domain.

Second, although our social media dataset has been treated as representing a unified domain, the cross-domain classification results show that there are evidently significant differences between the data from Twitter and from Reddit. Recall that it has already been pointed out that samples from Reddit are longer on average but vary much more in length (Section 3.3). While the transfer of training signals between Twitter and Reddit is certainly much better than from the Efcamdat to the (combined) social media domain, the results in Table 4.10 reveal that training on one and testing on the other is nonetheless significantly more difficult than in-domain prediction on the combined social media dataset. Furthermore, these results also suggest that Reddit is closer to Efcamdat than Twitter; prediction on the former benefits more from training on Efcamdat than prediction on the latter.

Finally, we present in Table 4.11 the results of mixing *all* of our data to form a single mixed-domain dataset and performing classification on a randomly chosen 75%/25%-train/test split of this set. As previously mentioned, the baseline is once again based on the majority class in the training data portion.

System + features	**Acc**	**Macro F1**
LogReg + word unigram + char n-grams	67.2	70.4
LogReg + word uni + bigrams + char n-grams	**67.5**	**70.9**
Majority class baseline	40.6	9.6

Table 4.11: Overview of the classification performance on the mixed-domain data based on a random 75%/25%-train/test split

In the fully mixed setting, in which samples are drawn from all three sources (Efcamdat, Twitter and Reddit), but in which the same, mixed domain applies to both the training and the test set, the classification performance is fairly reasonable and far superior to the baseline based on the majority class, which is `C1` in the full mixed dataset. The results lie between those of in-domain classification on Efcamdat and on social media data, which seems plausible. Performing level prediction on the single, mixed-domain dataset will be taken up again in the next chapter, which explores neural models in a multi-task learning setting, applied to this task.

[16] The size of our full Efcamdat dataset (Section 3.4)

4.4 Predictive Features in Each Domain

To gain a better understanding of our datasets drawn from and representing different domains, we printed out and list in the tables to follow the five most predictive features for each class in the Efcamdat, the combined social media, the Twitter, and the Reddit datasets. They were obtained by fitting a model[17] on a given dataset and subsequently extracting the five features with the highest weight coefficients.

In all of the below feature lists we use `[c]` to indicate a character n-gram feature and `[w]` a word n-gram feature. Moreover, all features per class are given in descending order, starting with the most predictive.

Efcamdat	
A1	`'m [c], i'm[c], i'm [c], hi[w], .  be[c]`
A2	`rie[c], leter[c], eter r[c], rien[c], rien [c]`
B1	`u [c], t a te[c], u c[c], u can[c], u ca[c]`
B2	`rtl[c], hortl[c], hortly[c], shortly[w], ortly[c]`
C1	`tyl[c], ortyl[c], shorty[c], hortyl[c], shortyl[w]`
C2	`robot[c], robo[c], robot[w], robo[c], obo[c]`

Table 4.12: Top 5 predictive features for each class in the EFcamdat data

Table 4.12 shows the five most predictive features by class for the Efcamdat dataset. They reveal to some extent the influence of topics on the Efcamdat dataset: For example, the top predictive features for level A1 are clearly related to phrases about social introductions, such as *Hi! I'm ...*. This is unsurprising, given that in a controlled foreign language learning context topics for beginners mostly relate to introducing oneself. Recall that Level 1 of the *EnglishTown/English Live* curriculum centres on, among others "Introducing yourself by email" and "Writing an online profile" (Section 3.4). This offers a plausible explanation for the misclassification of large portions of higher-level social media samples to level A1 (Table 4.8): Presumably, on social media, users of all proficiency levels will comment on events related to themselves, on their "current status", on what is on their mind etc. Thelwall (2009)'s examination of MySpace comments reveals highly pervasive usage of personal pronouns like *I* in those comments. When users' writings contain phrases like *I'm ...*, the Efcamdat-trained model is easily misled by them to classify the samples as belonging to level A1.

Table 4.13 shows the top predictive features in the combined social media dataset (i.e. Twitter and Reddit). Notice that the individual, isolated occurrences of "e" almost certainly represent emoticons, which have been normalised to "e" in the pre-processing step (Section 4.2). These, combined with the presence of some non-alphanumerical

[17] The SVM or the logistic regressor. Generally, the top predictive features for each class largely remained the same for all models we tested.

<table>
<tr><th colspan="2"><i>Social Media</i></th></tr>
<tr><td><b>A1</b></td><td>acts e[c], late e[c], cts e[c], orrow[c], rrow[c]</td></tr>
<tr><td><b>A2</b></td><td>why[c], why[w], feas[w], cutee[c], feas[c]</td></tr>
<tr><td><b>B1</b></td><td>yo [c], od e e[c], te[w], non.[c], sting[c]</td></tr>
<tr><td><b>B2</b></td><td>yo[w], oc[w], *of[c], non[w], eat[word]</td></tr>
<tr><td><b>C1</b></td><td>|| [c], for n[c], f god[c], e you.[c], w it e[c]</td></tr>
<tr><td><b>C2</b></td><td>cj[w], l so b[c], u you[c], tf[w], ht e e[c]</td></tr>
</table>

Table 4.13: Top 5 predictive features for each class in the combined social media data

features like || reflect the noisier nature of the data and that many predictive features are specific to the social media domain.

<table>
<tr><th colspan="2"><i>Twitter</i></th></tr>
<tr><td><b>A1</b></td><td>fr[w], mf[c], hoe[c], hoe[c], e e[c]</td></tr>
<tr><td><b>A2</b></td><td>of*[c], o 1[c], sonic[w], yo 1[c], feas[w]</td></tr>
<tr><td><b>B1</b></td><td>??[c], ugh[w], wig[c], wig[c], wig[w]</td></tr>
<tr><td><b>B2</b></td><td>^^[c], yo[w], -- [c], non[w], *-*[c]</td></tr>
<tr><td><b>C1</b></td><td>||[c], // [c] , ...[c], iked a[c], dan[w]</td></tr>
<tr><td><b>C2</b></td><td>♡[c], ...[c], ! e[c], esc[c], ♡e[c]</td></tr>
</table>

Table 4.14: Top 5 predictive features for each class in the Twitter subset of the social media data

<table>
<tr><th colspan="2"><i>Reddit</i></th></tr>
<tr><td><b>A1</b></td><td>e p[c], try[w], try[c], h a1[c], sh a1[c]</td></tr>
<tr><td><b>A2</b></td><td>ack[c], > [c], , and [c], , and[c], , s[c]</td></tr>
<tr><td><b>B1</b></td><td>20[c], franc[c], i w[c], fran[c], france[w]</td></tr>
<tr><td><b>B2</b></td><td>’s [c], n’t[c], ’t [c], n’t[c], ? [c]</td></tr>
<tr><td><b>C1</b></td><td>tai[c], l i[c], & [c], nt a[c], lol[c]</td></tr>
<tr><td><b>C2</b></td><td>t. [c], , b[c], , and[c], ted[c], e to t[c]</td></tr>
</table>

Table 4.15: Top 5 predictive features for each class in the Reddit subset of the social media data

Finally, Table 4.14 and Table 4.15 list the most predictive features in the Twitter versus the Reddit data as separate datasets. They again underline some differences between the two domains: Twitter seems to be characterised more by special characters and emoticons and appears to be closer to noisier web language than Reddit. This possibly explains why training signals from the Efcamdat domain transfer to it less successfully than to Reddit (Section 4.10).

4.5 Dealing with Ranked Nature of Class Labels

One important point only mentioned in passing in the previous discussions is the ordered nature of our six labels. Recall that the six labels form a *scale* in the CEFR scheme. Thus, the task does not simply deal with data of *nominal*, but *ordered* categories. Based on the traditional methods of computing precision, recall and the F1-measure for each class, which we did in order to obtain the macro-average F1 measures reported above, we effectively disregard the fact that misclassifying a C1 sample as A2, for instance, shows worse performance than misclassifying it as B2. On the other hand, since the data are not of *continuous* categories either, a linear regression model is not appropriate. Therefore, as an alternative to using a different *model*, a different *evaluation* method can be considered which does take the ordered nature of the labels into account. Hence, this section details our exploration of a *customised* version of the traditional metrics precision, recall and F1-measure. It is a more lenient adaptation which, put bluntly, also rewards the model if it mispredicts a sample but the predicted level is close to the gold-standard level on the proficiency scale.

4.5.1 Methods

To recall, the standard metrics *precision* and *recall*, which have been used in the previous sections, are defined as follows for each class C:

$$\textbf{Precision} = \frac{True\,Positives}{True\,Positives + False\,Positives} \tag{4.1}$$

$$\textbf{Recall} = \frac{True\,Positives}{True\,Positives + False\,Negatives} \tag{4.2}$$

where

- True Positives (TP) are: the number of samples which are *truly* from class C and *correctly* classified as such by the model
- False Positives (FP) are: the number of samples *not* from C but *falsely* classified as such by the model
- False Negatives (FN) are: the number of samples *truly* from class C but *falsely* classified as something else by the model

True Negatives (FN), i.e. the number of samples *not* from class C and *correctly* classified as something else by the model, are irrelevant here. Overall, precision indicates what percentage of samples which the model believes to be members of a given class C truly are members of C; recall indicates what percentage of samples which are truly members of C have been identified as such by the model.

The F1-measure is defined as the harmonic mean between precision and recall and computed as follows:

$$\textbf{F1} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.3}$$

Our customisation of these metrics target precision and recall, more specifically, the collection of the numbers of TP, FP and FN. The core idea is as follows:

In the standard method, given a sample for which the gold-standard and the predicted class labels differ, the sample entails one addition to FP of the predicted class as well as to FN of the gold-standard. Conversely, a sample for which the gold-standard class is identical to the predicted class adds one to TP of that class. In this treatment, correctness of class assignment is all-or-nothing, either coinciding with the gold-label (correct) or differing from it (false). In our alterations, we employ a nuanced approach to the correctness of class assignment based on the *distance* between the gold-standard and the predicted class on the class label scale. Thus, if a sample is labelled such that the predicted class is different from but close to the gold-standard, it nevertheless contributes a degree of correctness to TP of the predicted class. Conversely, it also adds a smaller degree of falseness to FP of the predicted and FN of the gold-standard class than in the standard method. More concretely, for each sample, hence each pair of gold versus predicted class label, we obtain

- a *correctness* score between 0 and 1, indicating the degree to which the predicted label is correct, judged against the gold, and
- a *falseness* score, indicating the reverse and computed as complementary to the correctness score, hence as $1 - correctness$

Where gold and predicted are identical, the correctness score will be 1 and falseness 0. Then, we add

- the *correctness* value to TP of the predicted class,
- the *falseness* value to both FP of the predicted class and to FN of the gold-standard class.

How to set the correctness score in relation to the proximity between the gold and the predicted labels is essentially a matter of choice. In our case, we chose to set correctness as

- **1** if the distance between gold and predicted is 0 (i.e. if they are identical)
- **0.6** if the distance between them is 1 (i.e. if they are adjacent on the label scale)
- **0.3** if the distance is 2 (i.e. if they are separated by one label)
- **0** if the distance is greater than 2.

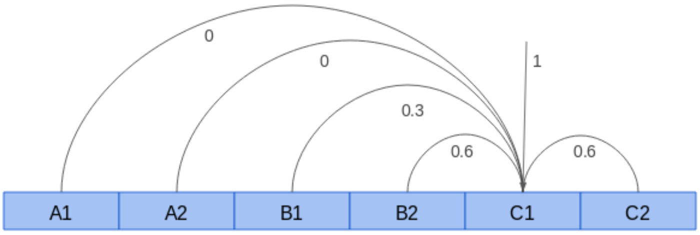


FIGURE 4.1: Visualisation of the correctness scores which hold between class C1 and its neighbours

Figure 4.1 visualises an example of these heuristics, with the arcs showing the correctness value which would hold between class `C1` and each of the other labels if they formed a pair of gold versus predicted labels. As already mentioned, should the gold and the predicted labels be identical, or if they are more than two labels apart from each other such that the prediction must be considered "completely false", the customised metrics will have the same effect as the standard ones.

The changes described above affect the values TP, FP and FN for each class and thereby also the precision and recall calculated. The F1-measure is based on the harmonic mean between the (now altered) precision and recall values, as in the standard case. To illustrate an application of our customised metrics, regard the following toy case consisting of 20 samples with their gold (G) versus predicted (P) labels:

$labels_G$ = [A2, B2, B2, B2, C1, B2, A2, C1, C1, C2, A1, A1, B2, C2, C1, C1, A2, B2, A2, C1]

$labels_P$ = [A2, B1, C2, C1, B1, B2, A1, A1, A2, C2, A2, B1, B2, A1, A1, B2, B2, A1, A2, B1]

Table 4.16 contrasts the values given by the standard classification metrics and our customised adaptation to this toy dataset.

	Standard			**Customised**		
	Precision	Recall	F1	Precision	Recall	F1
A1	0.0	0.0	0.0	37.5	33.3	35.3
A2	50.0	50.0	50.0	57.5	74.2	64.8
B1	0.0	0.0	0.0	45.0	39.1	41.9
B2	66.7	28.6	40.0	66.7	43.5	52.6
C1	0.0	0.0	0.0	37.5	35.7	36.6
C2	33.3	1.0	50.0	43.3	1.0	60.5

TABLE 4.16: Values from the standard vs. the customised precision, recall and F1 metrics for toy dataset

In the case of class `A1`, for instance, the standard evaluation metrics assigns 0 to both precision and recall and thereby also F1, given that there is not a single instance in the toy dataset in which `A1` is correctly assigned. However, there is an instance in which a true `A1`-sample has been assigned to class `A2` and another where a true `A2`-sample has been assigned to `A1`. Therefore, under usage of the customised metrics, the system still receives some measure of positive evaluation for its performance on class A1 since it has misclassified into near neighbours of the true class.

Overall, the figures outputted by our altered evaluation metrics are bound to be higher than those of the standard ones, and it would be pointless to compare the performance of different models with each other where some use the standard metrics and others the customised. However, given a model, we can, for instance, compare if it performs similarly on all six classes, judged using the standard versus the more lenient customised metrics. We can also use the same customised metrics for classification in different domains and examine if the difference in performance is observed to the same extent once the allowance is made for "minor" classification errors in the form of misclassification into nearby classes on the scale.

Concretely, the following two experiments were conducted:

1. We ran our best system for in-domain classification on the social media set, i.e. the logistic regression model with word uni and bigrams and character n-grams in the range of 3 to 6, and evaluated the results with the standard as contrasted to the customised metrics. A random train/test-split of 75%/25% was again used.

2. We again ran the cross-domain logistic regression model (with word unigrams and character n-grams) which trained on mixed-domain data and tested on 5,000 samples from the social media dataset (i.e. the top-most model in Table 4.10), this time, however, evaluated using the customised metrics. We then ran the same model in an in-domain run, where we tested on the same 5,000 samples but trained only on the remaining samples from the social media set. This was then also evaluated with the customised metrics. Recall from the previous findings and discussion that training on the mixed-domain dataset, with the 28,029 Efcamdat samples as extra training data, hardly yielded any improvement to level prediction in the social media domain (Section 4.3.2). This experiment now investigated this more closely by a) having both runs test on the same 5,000 test samples and by b) using the more lenient, "order-sensitive" evaluation metrics.

4.5.2 Results and Discussion

Table 4.17 contrasts the detailed results of the first experiment, i.e. the in-domain classification on the social media set, using the standard metrics on the one hand and our customised metrics on the other hand. Furthermore, Figure 4.2 visualises the differences with respect to the F1-measure for each proficiency level under the two evaluation schemes.

	Standard			Customised		
	Precision	Recall	F1	Precision	Recall	F1
A1	88.6	31.7	46.7	89.0	31.9	46.9
A2	47.4	8.8	14.8	54.2	10.5	17.6
B1	62.5	44.2	51.8	72.4	54.6	62.3
B2	59.0	37.0	45.5	79.6	61.6	69.5
C1	64.8	75.9	69.9	83.1	90.2	86.5
C2	58.1	59.8	58.9	76.8	80.9	78.8

TABLE 4.17: Standard vs. customised evaluation for in-domain classification on the social media dataset using the best logistic regression model

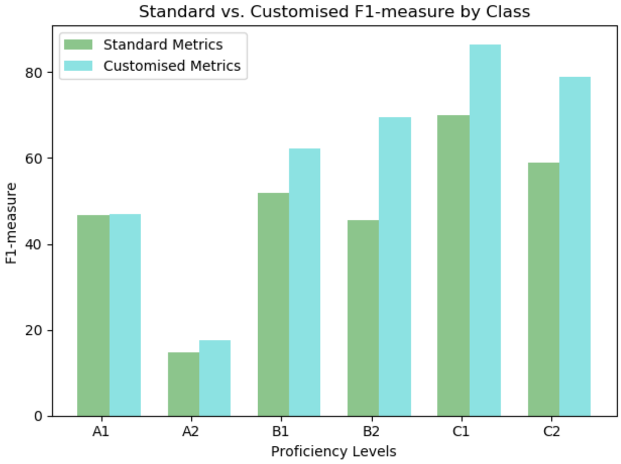


FIGURE 4.2: Bar chart for results for in-domain classification on social media dataset using logistic regressor: F1-measures by class based on standard vs. customised metrics

We see in these results that despite using the more lenient customised evaluation, performance in the classes `A1` and `A2` remain extremely poor. This shows that misclassification in these classes is largely not errors in fine-grained distinction between similar proficiency levels. Rather, the classifier is usually "utterly" wrong in that it confuses samples from these low levels to levels on the other end of the proficiency scale. This is in accordance with the observations from the confusion matrix in Table 4.4. However, in the other four levels, particularly B2, C1 and C2, usage of the customised metrics clearly raises their respective F1-measures, indicating that in a considerable amount of cases, while the system misclassies samples from these classes, they do place them in the generally correct area on the proficiency scale.

In Table 4.18 we contrast the detailed results of the second experiment, wherein we contrast two model runs tested on the same 5,000 samples of social media data. In one run ("No mixed-domain"), the model trained only on those social media samples not among the 5,000 test samples; in the other ("With mixed-domain"), it trained on these and all Efcamdat samples *in addition*, hence on a larger, mixed-domain dataset. Figure 4.3 visualises the F1-measures by class, contrasting the two runs.

	No mixed-domain			**With mixed-domain**		
	Precision	Recall	F1	Precision	Recall	F1
A1	82.6	39.3	53.2	87.5	43.4	58.0
A2	69.0	16.8	27.1	59.0	13.7	22.2
B1	73.2	58.0	64.7	74.4	57.2	64.7
B2	76.9	65.1	70.5	83.3	58.0	68.4
C1	84.0	90.5	87.1	82.6	93.1	87.5
C2	78.5	80.4	79.5	81.0	78.3	79.6

Table 4.18: Classification of social media data without and with training on extra Efcamdat data, evaluated using customised metrics

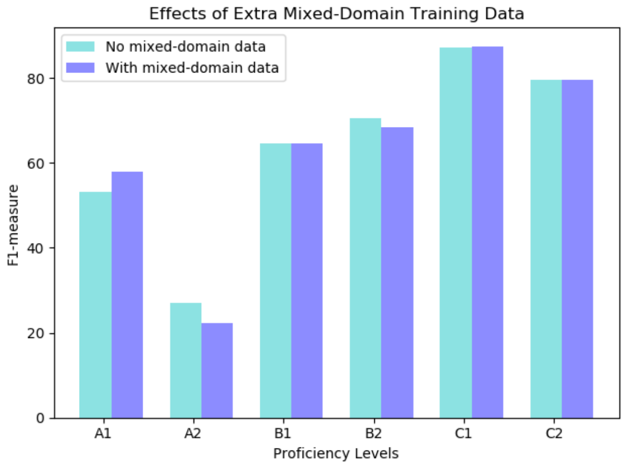


Figure 4.3: Bar chart for results for classification on social media dataset excluding and including extra training data from Efcamdat

These results further corroborate our earlier finding regarding the addition of the 28,029 Efcamdat samples as additional training data for a model tested on the social media: They are not useful. Figure 4.3 clearly shows no overall benefit of adding the extra data from a foreign domain. Given that these figures have been obtained by using our customised metrics, this means that the addition of the Efcamdat training data not only fails to yield more (fully) correct predictions on the social media test set, it does not even allow the model to "reduce the gravity of its errors" and place incorrectly classified samples closer to the gold label.

4.6 Chapter Conclusion

This chapter discussed our main experiments of making proficiency level predictions using two traditional machine learning systems, the SVM and the logistic regression model, and various features. In all cases, simple word and character n-grams proved to be the most useful. We looked at in-domain classification in both the Efcamdat and the social media datasets, respectively, and cross-domain classification wherein we attempted to use training signals from the former to inform the latter. The following findings have been gained: Rather unsurprisingly, classification can be done very well on the Efcamdat data, where the co-occurrence of certain topics with certain levels further made the task even easier. However, despite the noisy nature in the social media samples, level prediction is also possible on these data to some extent, at least when models are trained on data from the same domain. The accuracy and macro-average F1 measure reach up to 63.4% and 47.5%, respectively, which clearly beats the baseline. We found large differences between the performance on the six different levels, which reflect the impact of the highly skewed class distribution in the dataset. Finally, while in-domain classification on the social media data was possible, prediction in the cross-domain setting proved to be futile. Training signals from Efcamdat were revealed as hardly useful at all for making predictions in the social media domain.

Chapter 5

Multi-Task Learning on Mixed-Domain Dataset

The results from the previous chapter have shown, among others, that a) despite the noisy nature of social media data, prediction of the writers' levels of English is indeed feasible to some extent, and that b) it is not possible to use clean data from the language teaching context as training data for this purpose, the two domains being too different to inform each other in a cross-domain setting.

In a follow-up experiment, we took a closer look at making proficiency level predictions on the joint *mixed* dataset, which puts together the data from all three sources (Twitter, Reddit and Efcamdat). Given the difference between the datasets, we examined if level prediction for individual samples would benefit from a multi-task learning setting in which we introduced an auxiliary task aimed at predicting the dataset that the sample originates from. The experiment is intended as a simple pilot study introducing multi-task learning to proficiency level prediction to see whether it could be an interesting method for further research. An in-depth analysis of how much representation the two prediction tasks should share, to what extent the auxiliary task is recognised as being related to the main task etc. would be beyond the scope of this project.

In the sections to come, we first provide background information on multi-task learning (Section 5.1), then describe its integration into our present task (Section 5.2) and the two models we examined (Section 5.3), and describe our experimental set-up (Section 5.4). Section 5.5 presents and discusses our results.

5.1 Background: Multi-Task Learning (MTL)

In a seminal paper, Caruana (1997) makes a strong case in favour of MTL in machine learning, which he defines as "[S]haring what is learned by different tasks while tasks are trained in parallel", with the aim of better generalisation. He argues that many, if not most, real-world problems are multi-task rather than single-task problems and that, rather than breaking down problems into subtasks and training them separately, subtasks from the same domain gain from sharing information with each other and *learning in parallel*. This also means that, in MTL, a task of interest, the *main task*, can benefit from the introduction of parallel *auxiliary tasks*. The latter are not by themselves relevant but are useful to the main task by virtue of providing it with extra knowledge when they are trained in parallel. Caruana (1997) regards the training signals from the auxiliary tasks as useful *inductive bias* for the main. To name an example he reports, MTL can be effective in object recognition. When learning to predict the type of door shown to the machine learning system and the position of the door knob, it benefits from explicitly learning to predict a range of other properties of the door, such as its width or the position of its edges.

Caruana (1997) is inspired by previous work related, but not identical, to MTL: NetTalk (Sejnowski, 1987; Rosenberg & Sejnowski, 1986), an early neural network system that learns to "translate" written English texts into phonemes and word stresses to enable text-to-speech synthesising, already has multiple output units sharing the same learned representations. Although not dealing with parallel training using shared representations, Pratt (1992) reveals the feasibility of transferring learned weights from one network to a second network which is learning a related task and demonstrates the benefit of doing so (albeit mainly with regard to training speed). Similar advantages of knowledge transfer between related tasks are presented in Sharkey and Sharkey (1993).

Amongst others, Caruana (1997) identifies the following ways the auxiliary task can improve a system's performance on the main: Features shared by both tasks can be trained more effectively with more feed-back (from both sets of gold-standard labels), hence more training signals. Moreover, learning the auxiliary task may allow the system to "discover" features which are in fact useful to both tasks but which would not be recognised as useful to the main task in a single-task setting.

In recent years, MTL is increasingly being adopted by the NLP community: Luong, Le, Sutskever, Vinyals, and Kaiser (2015) examine MTL in a sequence-to-sequence learning context and show that neural machine translation benefits from the addition of the parallel tasks of syntactic parsing and image caption generation. In Klerke, Goldberg, and Søgaard (2016), sentence compression, a form of sentence simplification, is shown to be aided by parallel learning of reader gaze prediction, where intuitively, sections of the sentence that invite longer gaze fixation are also candidates for compression. On the other hand, Alonso and Plank (2016) examine the application of MTL to a range of semantic sequence prediction tasks, such as semantic role-labelling and named entity recognition, and find that they do not generally gain from the addition of auxiliary tasks such as dependency parsing or part-of-speech prediction. Bingel and Søgaard (2017)'s systematic investigation of under which conditions MTL works concludes that its success greatly relates to the learning curves of the tasks involved. They suggest that the best combination is one where the main task's learning by itself quickly becomes caught in local minima and reaches a plateau while the auxiliary task's learning does not plateau fast. In such cases, the auxiliary can help the main task out of the local minimum.

5.2 MTL in English Level Prediction

In the present study, we explored the application of MTL to the joint set of all our data, from both the social media and the language teaching context. The **main task** remains the automatic classification of writers' level of English as a foreign language on a per-sample basis, with the CEFR levels A1 to C2 as the possible classes. As an **auxiliary task**, we introduced the task of automatically predicting, for any given sample, the dataset which it was drawn from. As the previous chapter has shown, within social media, there appear to be sizeable differences between samples from Twitter and Reddit such that prediction across these two datasets is largely not successful. Therefore, with respect to the auxiliary task, we not only distinguished between the controlled language learning domain and the social media domain, but also between the two different sources of social media data, making the auxiliary task a 3-way classification problem.

Hence, the main task involved the six classes `A1`, `A2`, `B1`, `B2`, `C1` and `C2`, and the auxiliary task the three classes `Twitter`, `Reddit` and `Efcamdat`. Tables 5.1 and 5.2 give an overview of the class distributions in the joint dataset with respect to both the main and the auxiliary task. The Twitter dataset is by far the largest in terms of number of samples (although they are shorter, see Section 3.3). Level distribution is again skewed towards the higher levels, especially C1, since this is the trend in the largest dataset, i.e. the Twitter dataset (Section 3.1).

Level	A1	A2	B1	B2	C1	C2	**Total**
Samples	6,928	7,296	17,189	20,603	59,125	35,026	**146,167**

Table 5.1: Distribution of the six main task classes in the joint dataset

Data source	Twitter	Reddit	Efcamdat	**Total**
Samples	107,767	10,371	28,029	**146,167**

Table 5.2: Distribution of the three auxiliary task classes in the joint dataset

The intuition behind our MTL set-up is as follows: Given the differences between the datasets, it is possible that certain features which are predictive of the origin of given samples are also useful for improving level prediction on this mixed, joint dataset. However, such features might not be recognised as beneficial when level prediction is the *only* task being learned. In other words, by introducing the auxiliary task of data origin classification, we force the system to explicitly learn the differences between the three datasets while predicting proficiency levels on all of them.

Possibly the most straight-forward method of doing MTL is the use of neural networks, where all tasks being trained share all hidden representations and where each task simply has its distinct layer of output nodes. This is in accordance with Caruana (1997)'s observations (although he also sees ways of incorporating MTL into non-neural algorithms such as k-nearest neighbour). Therefore, we also conducted our experiments using basic versions of two neural architectures: the (bidirectional) Long Short Term Memory (Bi-LSTM), first proposed by Hochreiter and Schmidhuber (1997), and the Convolutional Neural Network (CNN), which dates back to Fukushima (1979, 2013) (see Schmidhuber (2015) for an overview). In recent years, both types of networks have become popular in various NLP applications. Amongst others, LSTMs and Bi-LSTMs have been used in natural language inference tasks (Nangia, Williams, Lazaridou, & Bowman, 2017), POS-tagging (Plank, Søgaard, & Goldberg, 2016) and neural machine translation (Bahdanau, Cho, & Bengio, 2014); CNNs have tranditionally been applied to vision-related tasks like hand writing recognition (Wu, Fan, He, Sun, & Naoi, 2014) and image captioning (Xu et al., 2015) but have also been shown to be successful in sentence classification tasks (Kim, 2014; Gambäck & Sikdar, 2017). Discussions of these models as such are beyond the scope of this thesis.

Notice that the main goal of our MTL-related experiments was to assess whether or not MTL including the auxiliary task would perform better on the main task, i.e. level prediction, than the main task as a single task. The goal was not achieving the best-possible level prediction results. Therefore, we did not attempt to improve either neural model we used by optimising hyper-parameter selection or increasing their complexity, but focused on contrasting their performance in the MTL setting with that in the single-task setting.

5.3 Architecture of Models

5.3.1 Bi-LSTM

In our simple Bi-LSTM model we use softmax-classification on top of the last hidden states of the forward and the backward LSTM. The overall architecture of the *single-task* model is as follows:

Embeddings NLTK's (Bird & Loper, 2004)'s word tokenizer is used to turn each input sample into a list of word tokens. We then represent all individual words as **64 dimensional** embeddings, randomly initialised and trained with the whole model.

Forward and Backword LSTM We feed the embedding representations of the sample into a standard Bi-LSTM architecture: The forward LSTM processes the sample in the regular order, reading in one word representation at each time step and outputting for each word a **64 dimensional** hidden state. The last hidden state, $\overrightarrow{h}_{Last}$, is taken as the representation of the full sample in the forward pass. The backward LSTM performs the same but reads in the sample in the reversed order, producing $\overleftarrow{h}_{Last}$, the sample representation in the backward pass. These two 64 dimensional last hidden states are then concatenated to yield the 128 dimensional $\overrightarrow{h}_{Final}$, the final vector representation of the full sample, hence:

$$\overrightarrow{h}_{Final} = \overrightarrow{h}_{Last} \oplus \overleftarrow{h}_{Last} \tag{5.1}$$

where $\oplus$ is the concatenation operator.

Softmax Finally, $\overrightarrow{h}_{Final}$ is passed to a softmax layer with a bias term for classification, which outputs probability scores for each of the six main task classes.

The only difference between the above *single-task* architecture and the *multi-task* one is that in the MTL setting, the sample representation $\overrightarrow{h}_{Final}$ is also fed to an additional, separate softmax layer for the 3-way classification of the auxiliary task. Figures 5.1 and 5.2 visualise the single and the multi-task Bi-LSTM models, respectively.

5.3.2 CNN

Our CNN is a simple model with softmax-classification on top of a single convolution layer, itself built on top of an embedding layer. In greater detail:

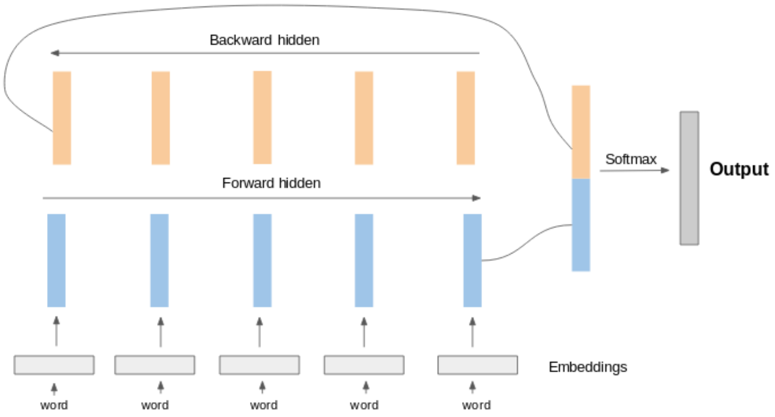


Figure 5.1: Bi-LSTM model in the *single-task* setting

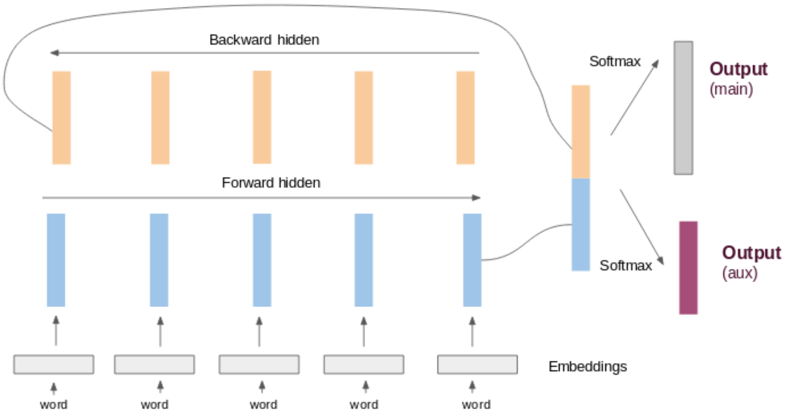


Figure 5.2: Bi-LSTM model in the *MTL* setting

Embedding The embedding layer is the same as that in the Bi-LSTM, with **64 dimensional** word embeddings generated "on the fly" for all word tokens of the input sample.

Convolution At the convolution layer with a bias term, all word embeddings of the sample are concatenated along axis 1, creating a matrix of size embedding dimensions (i.e. 64) by the length of the sample. A single filter moves over this representation, using the following CNN hyper-parameters:

- `stride`: 1
- `filter size`: 64
- `window size`: 3

This effectively means that we regard each possible trigram within the sample and extract from it a 64 dimensional feature vector through convolution. The resulting feature map then undergoes max-pooling to yield a single feature vector representing the full sample. To this, we further apply a non-linear transformation using `ReLU` to produce the final sample representation.

Softmax Once again identical to the Bi-LSTM, the vector representation of the full sample is then fed to the softmax function with a bias term for classification. Again, in the single-task setting, there is only one such output layer classifying into the six levels, while in the MTL one, there is an output layer in addition for the three-way auxiliary classification task.

Figures 5.3 and 5.4 show the single and the multi-tasl CNN models, respectively.

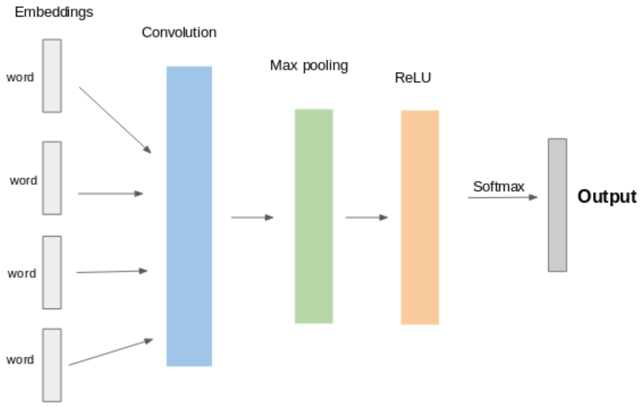


FIGURE 5.3: CNN model in the *single-task* setting

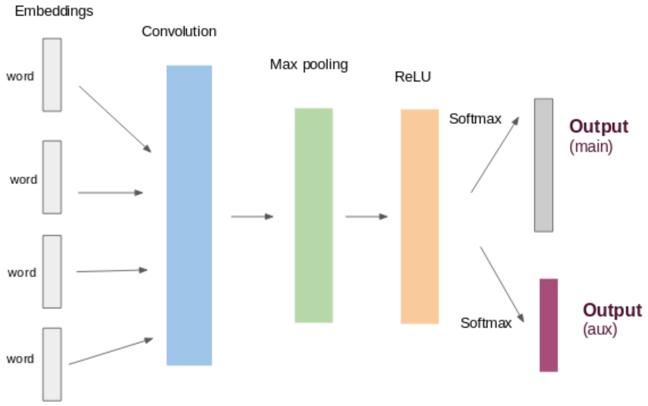


FIGURE 5.4: CNN model in the *MTL* setting

5.3.3 Loss Calculation

Loss is calculated in the same manner for both neural models. In either case, we use categorical cross-entropy to calculate the loss L, hence:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} 1_{y_i \in C_c} log(p_{model}[y_i \in C_c]) \tag{5.2}$$

where N is the number of samples and C the number of possible classes, $1_{y_i \in C_c}$ is the indicator function for the i-th sample belonging to the c-th class, and $log(p_{model}[y_i \in C_c])$ is the probability which the model assigns to the case of the i-th sample belonging to the c-th class.

In the MTL setting, this is done separately for both tasks. We then sum the loss from the main and the auxiliary task, controlling, however, the weighting of the two. In a similar manner to Mou et al. (2016), the total loss L_{Total} is calculated as

$$L_{Total} = \lambda\, L_{main} + (1 - \lambda)\, L_{aux} \tag{5.3}$$

where L_{main} and L_{aux} are respectively the individually calculated losses for the main and the auxiliary task and λ controls to what extend we place more (or less) weight on the main than on the auxiliary task; the larger the λ-value, the higher the importance of the main task and its training signals.

5.4 Experimental Set-up

All models in our experiments were implemented with the Python neural network toolkit DyNet (Neubig et al., 2017)[1], using their AdamTrainer[2] which incorporates optimisation with Adam (Kingma & Ba, 2014). In all cases, the input samples were lightly preprocessed by removal of control characters and detectable emoticons in the same manner as described in Chapter 4.

We held out 25% of the full dataset for testing and further split the the remaing data into a training and a dev set of 80% and 20%, respectively. We planned to train our models for ten epochs, i.e. ten iterations through the whole training set. However, single-task classification results on the dev set showed that overfitting already appeared to set in after three. Therefore, in the below section we report on the classification results by

[1] The implementation of both models is based on Graham Neubig's example implementations made available at https://github.com/neubig/nn4nlp-code

[2] http://dynet.readthedocs.io/en/latest/optimizers.html

all models on the held-out test data after *three* epochs of training. For these runs, we attached the dev dataset back to the train set so that we trained on a total of 109,625 samples and tested on the fixed test set of 36,542 samples.

With respect to the weighting of the main task loss versus the auxiliary task loss, we experimented with the following values for the λ-parameter:

$$\lambda = \{0.5,\ 0.8,\ 0.95,\ 0.999\} \tag{5.4}$$

5.5 Results and Discussion

All of our experimental results are presented in terms of accuracy and macro F1. Table 5.3 shows the test set results of either neural system in the *single-task* setting; Table 5.4 gives an overview of the test results in the *MTL* setting under the four different values for the weighting parameter λ.

	Acc	**Macro F1**
Bi-LSTM	64.2	69.1
CNN	61.2	64.5

Table 5.3: Test set results for the Bi-LSTM and the CNN in the *single-task* setting

	Main		**Aux**	
	Acc	**Macro F1**	**Acc**	**Macro F1**
$\lambda = 0.5$				
Bi-LSTM	**65.8**	**69.7**	94.4	85.7
CNN	61.0	63.8	93.2	79.7
$\lambda = 0.8$				
Bi-LSTM	65.7	69.1	94.3	82.6
CNN	**62.4**	**65.0**	92.3	69.3
$\lambda = 0.95$				
Bi-LSTM	**66.8**	**70.3**	92.9	74.2
CNN	62.6	64.0	91.6	67.9
$\lambda = 0.999$				
Bi-LSTM	65.2	68.2	90.8	68.9
CNN	62.0	63.8	91.2	62.8

Table 5.4: Test set results for the Bi-LSTM and the CNN in the *multi-task* setting with varying values for λ; Conditions where the multi-task performs better than single-task setting in terms of both accuracy and macro F1 are highlighted

As demonstrated by these figures, in several cases the incorporation of the auxiliary task in an MTL setting benefits the main task of level classification in terms of accuracy. Under three conditions, highlighted in Table 5.4, improvement is seen in both accuracy and the macro-average F1-measure, thus clearly showing improvement over the single-task scores.

Regarding the relative weighting between the main and the auxiliary task, these results are not fully conclusive, particularly in the case of the Bi-LSTM. In the case of the CNN, MTL's benefit is most pronounced at $\lambda = 0.8$, which places significantly more weight on the loss of the main than the auxiliary task. This also holds for the Bi-LSTM to some extent, with the largest improvement over the single-task counterpart at $\lambda = 0.95$. Benefits are also seen at $\lambda = 0.8$, albeit only in terms of accuracy (and not macro F1). Unlike the CNN, however, the Bi-LSTM also benefits from MTL in the $\lambda = 0.5$ condition, which gives equal weight to the main and the auxiliary task. Overall, the Bi-LSTM is possibly slightly more susceptible to the benefits of MTL than the CNN, although this would need to be established by further systematic studies looking into a larger set of parameters. In either model, giving too much weight to the main task, with $\lambda = 0.999$, harms performance, likely since the training signals from the auxiliary task are too little to be of use.

With regard to the models' performance on the auxiliary task of predicting a sample's source dataset, the coherent picture is that the less weight on the auxiliary task, the lower the scores from either neural model. This appears plausible.

5.6 Chapter Conclusion

This chapter does not (and does not intend to) offer a *deep* understanding of MTL in the prediction of proficiency levels on the mixed dataset, but intends to examine whether or not introducing the intuitively sound auxiliary task of dataset prediction is at all a viable option for improving level prediction in an MTL architecture. We implemented two neural models which are frequently used in NLP, viz. the Bi-LSTM and the CNN, and contrasted their performance in level prediction including and excluding the auxiliary task. Based on our experiments, we can clearly conclude that incorporation of the auxiliary task benefits level prediction, given the appropriate weighting of the main and the auxiliary task, and that MTL in such a setting merits further attention. Without complex model architectures and any parameter tuning, for both the Bi-LSTM and the CNN we obtained improvements over their single-task counterparts in terms of both classification accuracy and macro-average F1. Such an MTL set-up might well be applicable to other tasks where a model is required to learn from a mixed-domain dataset. More thorough studies into how to determine the optimal weighting scheme for the main and auxiliary task signals, how much representation all tasks should optimally share, how MTL will perform in more complex, deeper neural architectures etc. will shed more light on the benefits of MTL.

Chapter 6

Concluding Remarks

6.1 Summary and Main Findings

In this project, we performed automatic learner level prediction for English, using elicited data on the one hand and spontaneous data on the other. Given a written sample produced by some non-native speaker of English, the task was to predict the writer's level of English in terms of one of the six levels in the CEFR system. Our elicited data were drawn from the Efcamdat corpus, a learner corpus comprising short essays written by English learners as part of an online English course. In contrast, our spontaneous dataset was extracted from the social media platforms Twitter and Reddit, distantly supervised based on users' self-reported proficiency levels. We performed a series of classification experiments both *within* and *across* the two datasets as well as on their joint, mixed-domain dataset. In terms of classification methods, we experimented both with linear SVM and logistic regression, as well as with two neural systems in a multi-task learning setting.

Our main findings, in response to our core research questions (Chapter 1), are as follows: We see that it is indeed possible to obtain a set of distantly level-annotated data from social media, based on taking users' self-reported proficiency level to be the gold labels. While the annotation is undoubtedly noisy, it is dependable enough to be used for training supervised machine learning models with considerable success. It should be noted, however, that our search for users who make reliable proficiency level self-reports was not fully automatic but relied on much manual validation. Moreover, we do also find that using user-reported levels as distant labels entails an unbalanced level distribution in the dataset, seeing that self-reports for the lower levels A1 and A2 are understandably rare.

Our experiments show that, in spite of the difficulty posed by noise and sample brevity, automatic prediction of learner levels from social media data texts is possible to some extent. Using word unigrams and bigrams as well as character n-grams, our best in-domain logistic regression model achieved an accuracy score of **63.4%** and a macro-average F1 value of **47.5%**, which by far beats the baseline based on the training set majority class. Hence, there are clearly useful and exploitable training signals in data from the social media domain.

However, our findings suggest that level prediction in the social media domain do *not* benefit from training in the language learning domain, the latter being too different to inform the former. When training on the elicited Efcamdat data and testing on our spontaneous dataset collected from social media, system performance is disastrously bad. Direct transfer of level prediction training signals from the controlled language learning context to the undirected social media context seems not to be possible.

6.2 Future Directions

Finally, we outline a few possible directions for further research in the direction of automatically predicting learner levels based on spontaneous data:

A simple alteration to our experimental setting is to perform the prediction on a by-user instead of a by-sample basis. Assuming that most users will each be the authors of a set of multiple writings, by-user classification might be an easier task as there would be more data to base predictions on than a single writing sample, which in some cases can be little more than a short comment. It would also place the task closer to the field of author profiling.

If feasible, it would also be interesting to have a small portion of the social media data assessed and assigned to CEFR levels by human annotators (although this might be difficult). Our in-domain classification on the social media dataset suggest that our system is indeed learning to differentiate between proficiency levels. It could be rewarding to validate this finding on a set of human-labelled test set.

With regard to the collection of the dataset, As once again mentioned above, our social media dataset is highly unbalanced and heavily skewed towards the higher proficiency levels, especially the levels C1 and C2. This is a plausible consequence of the manner in which the dataset was created. Thus, future studies would ideally employ a second, complementary method of distant supervision to generate more training samples specifically for the lower levels.

Last but nor least, since our experiments found cross-domain classification to be unsuccessful, for one thing, a closer examination of the corpus characteristics and differences between the two domains could be of interest. We could statistically evaluate the n-gram distribution, the frequencies of function words versus content words, the mean length, concreteness and other properties of the words composing each dataset etc. In particular, it would be interesting to see if the features which Hawkins and Buttery (2010) and Crossley et al. (2012) identify as being predictive of learner levels vary to such an extent between the elicited and the spontaneous data domains that in a cross-domain context, they simply lose their value as predictors of proficiency level. In this context, exploration of domain transfer, an entire field in its own right, would certainly be of interest. Our own experiment involving multi-task learning is a small pilot experiment and has shown promising results. Certainly more insight on how relevant training signals can be passed between tasks and/or domains would benefit learner level prediction.

To our knowledge, our experiments are the first to look at applying (English) learner level prediction to social media texts. In the above, we have identified only a few possible continuations of the project. Certainly, much more in-depth research with regard to this task is still needed in the future.

References

Alexopoulou, T., Geertzen, J., Korhonen, A., & Meurers, D. (2015). Exploring big educational learner corpora for sla research: Perspectives on relative clauses. *International Journal of Learner Corpus Research*, *1*(1), 96–129.

Alonso, H. M., & Plank, B. (2016). When is multitask learning effective? semantic sequence prediction under varying data conditions. *arXiv preprint arXiv:1612.02251*.

Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, *4*(3).

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Basile, A., Dwyer, G., Medvedeva, M., Rawee, J., Haagsma, H., & Nissim, M. (2017). Is there life beyond n-grams? a simple svm-based author profiling system. *Cappellato et al.[13]*.

Bingel, J., & Søgaard, A. (2017). Identifying beneficial task relations for multi-task learning in deep neural networks. *arXiv preprint arXiv:1702.08303*.

Bird, S., & Loper, E. (2004). Nltk: the natural language toolkit. In *Proceedings of the acl 2004 on interactive poster and demonstration sessions* (p. 31).

Boyd, A., Hana, J., Nicolas, L., Meurers, D., Wisniewski, K., Abel, A., … Vettori, C. (2014). The merlin corpus: Learner language and the cefr. In *Lrec* (pp. 1281–1288).

Bröckling, M., Coquaz, V., Fanta, A., Langley, A., Munafò, M., Pütz, J., … Wazir, R. (2018, May). *Political Speech Project.* https://rania.shinyapps.io/PoliticalSpeechProject/.

Bryant, C., & Ng, H. T. (2015). How far are we from fully automatic high quality grammatical error correction? In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* (Vol. 1, pp. 697–707).

Burstein, J. (2003). The e-rater® scoring engine: Automated essay scoring with natural language processing.

Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M., Braden-Harder, L., & Harris, M. D. (1998). Automated scoring using a hybrid feature identification technique. In *Proceedings of the 17th international conference on computational linguistics-volume 1* (pp. 206–210).

Caruana, R. (1997). Multitask learning. *Machine learning*, *28*(1), 41–75.

Chierchia, G. (1998). Reference to kinds across language. *Natural language semantics*, *6*(4), 339–405.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Council of Europe. (2001). *Common european framework of reference for languages: learning, teaching, assessment.* Cambridge University Press.

Crossley, S. A., Greenfield, J., & McNamara, D. S. (2008). Assessing text readability using cognitively based indices. *Tesol Quarterly*, *42*(3), 475–493.

Crossley, S. A., Salsbury, T., & McNamara, D. S. (2012). Predicting the proficiency level of language learners using lexical indices. *Language Testing*, *29*(2), 243–263.

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*.

DeKeyser, R. M. (2005). What makes learning second-language grammar difficult? a review of issues. *Language learning*, *55*(S1), 1–25.

Del Vigna, F., Cimino, A., Dell'Orletta, F., Petrocchi, M., & Tesconi, M. (2017). Hate me, hate me not: Hate speech detection on facebook. *In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy*.

De Marneffe, M.-C., & Manning, C. D. (2008). *Stanford typed dependencies manual* (Tech. Rep.). Technical report, Stanford University.

Ekman, P. (1971). Universals and cultural differences in facial expressions of emotion. In *Nebraska symposium on motivation.*

Emmery, C., Chrupała, G., & Daelemans, W. (2017). Simple queries as distant labels for predicting gender on twitter. In *Proceedings of the 3rd workshop on noisy user-generated text* (pp. 50–55).

Fukushima, K. (1979). Neural network model for a mechanism of pattern recognition unaffected by shift in position-neocognitron. *IEICE Technical Report, A*, *62*(10), 658–665.

Fukushima, K. (2013). Artificial vision by multi-layered neural networks: Neocognitron and its advances. *Neural networks*, *37*, 103–119.

Gambäck, B., & Sikdar, U. K. (2017). Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online* (pp. 85–90).

Geertzen, J., Alexopoulou, T., & Korhonen, A. (2013). Automatic linguistic annotation of large scale l2 databases: The ef-cambridge open language database (efcamdat). In *Proceedings of the 31st second language research forum. somerville, ma: Cascadilla proceedings project.*

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., . . . Smith, N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies: Short papers - volume 2* (pp. 42–47). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=2002736.2002747

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, *1*(12).

Goot, R., Ljubešić, N., Matroos, I., Nissim, M., & Plank, B. (2018). Bleaching text: Abstract features for cross-lingual gender prediction. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)* (Vol. 2, pp. 383–389).

Hancke, J., & Meurers, D. (2013). Exploring cefr classification for german based on rich linguistic modeling. *Learner Corpus Research*, 54–56.

Hawkins, J. A., & Buttery, P. (2010). Criterial features in learner corpora: Theory and illustrations. *English Profile Journal*, *1*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Honnibal, M., & Montani, I. (2017). spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Huang, Y., Murakami, A., Alexopoulou, T., & Korhonen, A. (2018). Dependency parsing of learner english. *International Journal of Corpus Linguistics*, *23*(1), 28–54.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Phil. Trans. R. Soc. A*, *374*(2065), 20150202.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klerke, S., Goldberg, Y., & Søgaard, A. (2016). Improving sentence compression by learning to predict gaze. *arXiv preprint arXiv:1604.03357*.

Leacock, C., Chodorow, M., Gamon, M., & Tetreault, J. (2010). Automated grammatical error detection for language learners. *Synthesis lectures on human language technologies*, *3*(1), 1–134.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, *18*(17), 1-5. Retrieved from http://jmlr.org/papers/v18/16-365

Luong, M.-T., Le, Q. V., Sutskever, I., Vinyals, O., & Kaiser, L. (2015). Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, *19*(2), 313–330.

Master, P. (1997). The english article system: Acquisition, function. *System*, *25*(2), 215–232.

Mou, L., Meng, Z., Yan, R., Li, G., Xu, Y., Zhang, L., & Jin, Z. (2016). How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111*.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov, V. (2016). Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)* (pp. 1–18).

Nangia, N., Williams, A., Lazaridou, A., & Bowman, S. R. (2017). The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. *arXiv preprint arXiv:1707.08172*.

Neubig, G., Dyer, C., Goldberg, Y., Matthews, A., Ammar, W., Anastasopoulos, A., ... Yin, P. (2017). Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., & Bryant, C. (2014). The conll-2014 shared task on grammatical error correction. In *Proceedings of the eighteenth conference on computational natural language learning: Shared task* (pp. 1–14).

Nicholls, D. (2003). The cambridge learner corpus: Error coding and analysis for lexicography and elt. In *Proceedings of the corpus linguistics 2003 conference* (Vol. 16, pp. 572–581).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, *12*(Oct), 2825–2830.

Plank, B., Søgaard, A., & Goldberg, Y. (2016). Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *arXiv preprint arXiv:1604.05529*.

Pratt, L. Y. (1992). Non-literal transfer among neural network learners. *Colorado School of Mines: MCS-92-04*.

Purver, M., & Battersby, S. (2012). Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th conference of the european chapter of the association for computational linguistics* (pp. 482–491).

Rangel, F., Rosso, P., Chugur, I., Potthast, M., Trenkmann, M., Stein, B., ... Daelemans, W. (2014). Overview of the 2nd author profiling task at pan 2014. In *Clef 2014 evaluation labs and workshop working notes papers, sheffield, uk, 2014* (pp. 1–30).

Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., & Inches, G. (2013). Overview of the author profiling task at pan 2013. In *Clef conference on multilingual and multimodal information access evaluation* (pp. 352–365).

Rangel, F., Rosso, P., Montes-y Gómez, M., Potthast, M., & Stein, B. (2018). Overview of the 6th author profiling task at pan 2018: Multimodal gender identification in twitter. *Working Notes Papers of the CLEF*.

Rangel, F., Rosso, P., Potthast, M., & Stein, B. (2017). Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. *Working Notes Papers of the CLEF*.

Rangel, F., Rosso, P., Potthast, M., Stein, B., & Daelemans, W. (2015). Overview of the 3rd author profiling task at pan 2015. In *Clef* (p. 2015).

Rangel, F., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., & Stein, B. (2016). Overview of the 4th author profiling task at pan 2016: cross-genre evaluations. In *Working notes papers of the clef 2016 evaluation labs. ceur workshop proceedings/balog, krisztian [edit.]; et al.* (pp. 750–784).

Richardson, L. (2013). Beautiful soup. *Crummy: The Site*.

Rosenberg, C. R., & Sejnowski, T. J. (1986). The spacing effect on nettalk, a massively parallel network. In *Proceedings of the eighth annual conference of the cognitive science society* (pp. 72–89).

Rosenthal, S., Farra, N., & Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 502–518).

Ruppenhofer, J., Siegel, M., & Wiegand, M. (2018, March). *Guidelines for IGGSA Shared Task on the Identification of Offensive Language.* http://www.coli.uni-saarland.de/ miwieg/Germeval/guidelines-iggsa-shared.pdf.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

Schmidt, A., & Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural* *language processing for social media. association for computational linguistics, valencia, spain* (pp. 1–10).

Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval* (Vol. 39). Cambridge University Press.

Sejnowski, T. (1987). Net talk: A parallel network that learns to read aloud. *Complex Systems*, *1*, 145–168.

Sharkey, N. E., & Sharkey, A. J. (1993). Adaptive generalisation. *Artificial Intelligence Review*, *7*(5), 313–328.

Shuyo, N. (2010). *Language detection library for java.* Retrieved from http://code.google.com/p/language-detection/

Speer, R., Chin, J., Lin, A., Jewett, S., & Nathan, L. (2017, September). *Luminosoinsight/wordfreq: v1.7.* Retrieved from https://doi.org/10.5281/zenodo.998161 doi: 10.5281/zenodo.998161

Tack, A., François, T., Roekhaut, S., & Fairon, C. (2017). Human and automated cefr-based grading of short answers. In *Proceedings of the 12th workshop on innovative use of nlp for building educational applications* (pp. 169–179).

Thelwall, M. (2009). Myspace comments. *Online Information Review*, *33*(1), 58–76.

Vajjala, S., & Loo, K. (2014). Automatic cefr level prediction for estonian learner text. In *Proceedings of the third workshop on nlp for computer-assisted language learning* (pp. 113–127).

Wu, C., Fan, W., He, Y., Sun, J., & Naoi, S. (2014). Handwritten character recognition by alternately trained relaxation convolutional neural network. In *Frontiers in handwriting recognition (icfhr), 2014 14th international conference on* (pp. 291–296).

Xia, M., Kochmar, E., & Briscoe, T. (2016). Text readability assessment for second language learners. In *Proceedings of the 11th workshop on innovative use of nlp for building educational applications* (pp. 12–22).

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., … Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048–2057).

Yannakoudakis, H., Briscoe, T., & Medlock, B. (2011). A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1* (pp. 180–189).