Bilingual Transfer Learning for Online Product Classification

Erik Lehmann, András Simonyi
Frankfurt School of Finance
erik.lehmann91@gmail.com
andras.simonyi@gmail.com

Lukas Henkel, Jörn Franke
European Central Bank
lukas.henkel@ecb.europa.eu
jorn.franke@ecb.europa.eu

Abstract

Consumer Price Indices (CPIs) are one of the major statistics produced by Statistical Offices, and of crucial importance to Central Banks. Nowadays prices of many consumer goods can be obtained online, enabling a much more detailed measurement of inflation rates. One major challenge is to classify the variety of products from different shops and languages into the given statistical schema consisting of a complex multi-level classification hierarchy - the European Classification of Individual Consumption according to Purpose (ECOICOP) for European countries, since there is no model, mapping or labeled data available. We focus in our analysis on food, beverage and tobacco which account for 74 of the 258 ECOICOP categories and 19 % of the Euro Area inflation basket. In this paper we build a classifier on web scraped, hand-labeled product data from German retailers and transfer to French data using cross lingual word embeddings. We compare its performance against a classifier trained on single languages and a classifier with both languages trained jointly. Furthermore, we propose a pipeline to effectively create a data set with balanced labels using transferred predictions and active learning. In addition, we test how much data it takes to build a single language classifier from scratch and if there are benefits from multilingual training. Our proposed system reduces the time to complete the task by about two thirds.

1 Introduction

Consumer price inflation in the euro area is measured by the Harmonised Index of Consumer Prices (HICP), which is calculated based on a basket of goods and services. While prices for some goods, like energy prices, are easy to observe, prices for many product groups, like e.g. food, are often collected manually. This survey-based approach is relatively expensive and slow. Nowadays we have access to data on prices of individual products sold on the internet. This kind of data can help to improve the quality of the data and monitor it at a higher frequency. There are various initiatives working on the usage of alternative data sources for the calculation of price statistics, but these are mainly country-specific. Examples are the use of scanner data (Białek and Berkesewicz, 2020) and web scraped food data (Macias and Stelmasiak, 2019) to measure Polish inflation, the measurement of the CPI in Finland (Koskimäki and Ylä-Jarkko, 2003) or forecasting daily CPI in the United Kingdom (Powell et al., 2018). The billion prices project (Cavallo and Rigobon, 2016) extensively researched this topic and validated the usefulness by backtesting against traditional measured price indices. Furthermore, they use online prices also to answer macroeconomic research questions, like the verification of inflation statistics from Argentina or the review of effects from changes in US trade policy (Cavallo et al., 2019).

There is an important problem related to online price data, which is rarely addressed but always encountered in the process: the classification of millions of online products into categories. This is crucial for analysing the individual components underlying inflation as required by consumer price indexes. Those classification systems are rather complex with hundreds of categories within several layers of hierarchy. Usually experts in the specific classification system are needed and the classification cannot

Disclaimer: This paper solely expresses the opinion of the authors. Their views do not necessarily reflect those of the ECB.

Proceedings of the Workshop on Natural Language Processing in E-Commerce (EComNLP), pages 21–31
Barcelona, Spain (Online), Dec 12, 2020.

be done by people without this expertise. Additionally, the product set is not static. During our investigation of scraped web product data, we found out that products are phased out and new products appear frequently. Thus over time much more different products with the same or completely different characteristics appear than originally available and a lot of them disappear again, meaning that constant monitoring of the system is necessary.

Using supervised machine learning methods to automatize a classification task of this type often requires a large amount of labeled training data and labeling by hand is very time consuming as it needs to be done for every language individually. Moreover, all the texts (product names, descriptions and categories) have to be represented by numerical features or feature vectors, which leads into the field of natural language processing (NLP). Many of the previously mentioned initiatives have neglected a detailed classification or only offer very broad classifications. This makes them less suitable for analysing individual inflation components. Furthermore, they do not take into account the fact that product data can be available in different languages, which is especially important for the polyglot European market.

In this study we address how contemporary NLP and machine learning techniques can be used to automate the classification of online products, and what kind of effort is needed to build a model from scratch without having labeled training data. The underlying classification system is the European Classification of Individual Consumption according to Purpose (ECOICOP) (European Union, 2016) used by the European Statistic offices and the European Union.

We found the vocabulary of the product texts to be very domain specific and pretrained language models only to cover parts. For this reason we have been working on ways to extend the vocabulary of existing models, which limited the methods we could use.

In particular, we look at ways of transferring knowledge contained in a classification model for product data in one language to another using cross-lingual word embeddings. In addition, we propose different neural architectures for monolingual training, bilingual transfer and bilingual training. We also report how much labeled data is needed for a decent model and compare the results of zero-shot bilingual transfer to bilingual training. On the basis of our findings we propose an active learning pipeline to create balanced training data in a target language. The goal is to build a tool-set for a multi-lingual collection of web-scraped product corpora to make product classification accessible in many languages.

2 Data

2.1 ECOICOP Classification System

Consumer inflation and CPI in the euro area are based on the ECOICOP schema. The classification schema has different hierarchies; we focus on the five-digit level, which consist of 258 different categories and is identical for all countries in the euro area. The 5-digit level distinguishes between, e.g., ”rice”, ”bread” or ”pasta products and couscous” which all belong to the category ”bread and cereals” at the four-digit level. The 5-digit level is currently used for the calculation of the CPI and inflation (Eurostat, 2020).

We limit our analysis to the following two two-digit categories and hierarchical subcategories up to the 5-digit level, because those match our scraped data from supermarket websites:

- food and non-alcoholic beverages (01.)
- alcoholic beverages and tobacco (02.)

Together they account for 74 of the 258 ECOICOP 5-digit categories and 19% of the euro area inflation basket. All other items are classified as non-food.

The task of COICOP classification of products cannot be delegated to anyone. The classification system is complex and requires expertise in food classification as the categories are described using a language/concepts that only statistic experts understand. Furthermore, expert classification ensures also more consistency in the classification of products. An example of an ECOICOP 5 digit category is the following:

- Food and non-alcoholic beverages (01.)

ECOICOP category	# of observations
9999 Non-Food	9,396
1171 Fresh or chilled vegetables [...]	1,182
2121 Wine from grapes	1,177
⋮	⋮
2123 Fortified wines	23
2202 Cigars	10
2112 Alcoholic soft drinks	8

Table 1: COICOP categories observations

- Food (01.1)
  - ∗ Bread and Cereals (01.1.1)
    - · Pizza and quiche (01.1.1.5)

2.2 Product Data

The product data is scraped from selected online shops of supermarkets in Germany, France and Belgium. The products sold by different supermarkets are similar, but each website provides different information or categorization. In particular, we use the product name and category given by the supermarket. For example, a product name from a German supermarket would be ”Eiweiß Toastbrötchen” and a supermarket category would be ”Lebensmittel / Frühstück / Brot / Brötchen”. Both contain very specialised terminology that are usually not found in many public corpora, especially brand names or special food descriptions. For example, the German word ”Zitronenglasur” (lemon glaze) does not exist currently in the German Wikipedia, but is commonly found in supermarkets. Furthermore, the supermarket category can usually not be mapped to COICOP. In addition, we extracted words from the product URLs as they often provide additional information.

For many products we have further information like product description, producer, brand, quantity or ingredients. However, due to their variability, especially across shops, we did not include them in our current research.

We do some basic preprocessing steps on the text including tokenizing, lowercasing and replacing special characters and numbers with the goal of reducing the vocabulary and making it more language independent. The texts from name, categorization and URL are concatenated with a separator token <sep> in between. The average German text consists of 19 tokens, while the average French text of 28 tokens. We classified 31,000 random products of eight German supermarkets and 22,000 products from two French supermarkets, two thirds by hand and one third rule based with manual validation. The appearance of products per category is imbalanced, with a mean of 419 products per class for the German data (see Table 1).

The French data is distributed similarly, as a difference we do not have tobacco products. The reason for this category imbalance is the products' imbalanced occurrence on the shop websites, i.e. the distribution corresponds to what is available in reality.

3 Related Work

Although there is interest for detailed inflation monitoring, only a small amount of research exists about classification into the ECOICOP or similar schemes for online product data. For example, the billion prices project emphasize the use of big data for inflation measurement (Cavallo and Rigobon (2016)). They collected web scraped data from large multi-channel retailers in the United States and built a Naive Bayes classifier on ”language-specific, hand-categorized items”. To our knowledge there is just a single study on using vector representations of words in the context of COICOP classification: Martindale et al. (2019) take web scraped clothing data and use a semi-supervised approach to create training labels for their products. They useed various methods to spread existing labels. They found them to be correctly classified in 70% of all cases.

Text classification. Text classification is one of the core NLP tasks with a huge number of applications (e.g., sentiment analysis, spam filtering, and intent detection in chatbots to mention some of the most important ones). Specifically, the classification of short texts that do not necessarily consist of a well-formed sentence or sentences is also a much-studied problem (Wang et al., 2017; Saha et al., 2019; Chen et al., 2019). State-of-the-art short text classification models typically consist of a deep neural architecture on top of a static word or subword embedding layer. While recurrent neural nets (RNNs) using long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) are often the first choice for sequential data like time series or texts, one-dimensional convolutional neural networks (CNNs) were found to outperform LSTM-based RNN variants in many settings, especially in the case of short or unstructured texts (Lee and Dernoncourt, 2016; Seo et al., 2020).

Transfer learning. Using representations learned for a supervised task on a large data set as a basis for building and training a model for a different data set is a frequently used method for addressing data scarcity problems. One of the first models widely used for this kind of transfer learning was the AlexNet convolutional image classifier by Krizhevsky et al. (2012). For the transfer the upper layers are cut off while lower layers get frozen and on top of them new layers are trained on the new task. Word embeddings are another example where representations trained on one task (to predict a word given its surrounding) are used for many other tasks. Originally, an important limitation of pre-trained word embeddings was their limited vocabulary, since they provided representations only for words that occurred in the training corpus. To overcome this limitation, Facebook AI Research additionally trained their embedding model on certain subwords occurring in the corpus as well (Bojanowski et al., 2017). The resulting framework (fastText) provides models pre-trained on Wikipedia for a large amount of languages. Klementiev et al. (2012) trained word embeddings jointly for English and German to obtain representations in one common vector space. Transferring knowledge from a classifier only trained on English data to a classifier for German data using these cross-lingual word vectors outperformed a simple translation approach. In the following section we use a common vector space for French and German to transfer the ECOICOP classification. A very successful approach for cross-lingual zero shot learning was introduced by Eriguchi et al. (2018). They used state-of-the-art transformer-based language models trained on multiple languages. Only the encoder part of the model was transferred, why the decoder was replaced with their own classification layer with impressive results in transferring the learned information between English and French. Earlier, Johnson et al. (2017) already showed that a single Neural Machine Translation model trained on multiple languages can generalize to some extent, allowing for zero shot prediction on unseen data.

4 Framework

We propose a CNN architecture (cf. Figure 2) on top of word vector representations exemplified in Figure 1.

4.1 Results

We start with a discussion of cross-lingual word embeddings and methods of extending the vocabulary to take into account the specialised vocabulary found in our product data. Then we describe the network architecture based on a convolutional neural network and the aforementioned embeddings. Finally, we present results obtained by using single language training, by using transfer learning from German to French, and by multilingual joint training of the model on both languages. The NLP research of the last few years has been dominated by the rise of the transformer neural architecture, also for multilingual problems (Johnson et al., 2017). For our specific problem the typically used pretrained transformer models were not suitable because there is no straightforward way to include domain specific vocabulary.

4.2 Cross-lingual Embeddings

Context. Word embeddings represent words in a high-dimensional vector space in which semantic relations between words correspond to geometric relationships. The embeddings are trained on the task

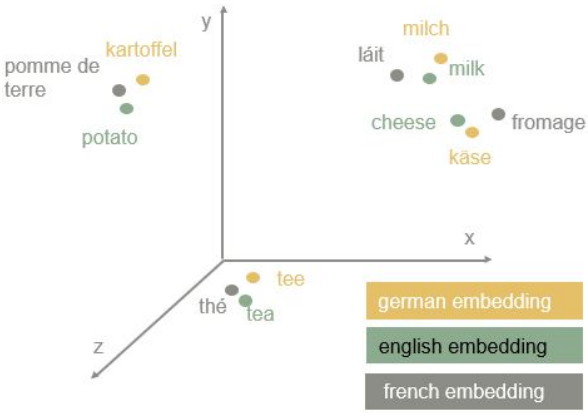

Figure 1: Artificial joint plot of different embedding models

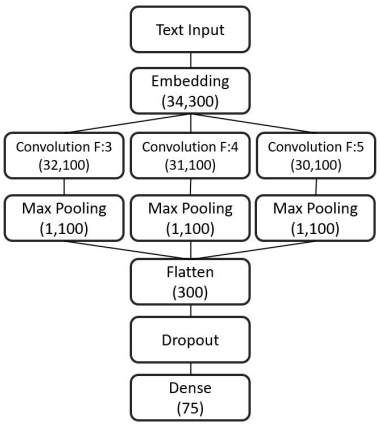

Figure 2: CNN architecture following the example of Kim (2014)

of predicting a word given its context, thereby, indirectly, a joint probability distribution of words is also learned. The learned vector representations capture both syntactic and semantic properties of words because they have to represent all information about words that is useful for the prediction task. As these syntactic and semantic properties are often similar across languages, word vectors from different languages can be aligned into one common space in which cross-lingual semantic and syntactic relations are also represented by geometric ones. The alignment is typically realized by applying a linear mapping between the two vector spaces (Mikolov et al., 2013). This mapping is often learned in a supervised manner by using a bilingual dictionary or parallel texts with the objective of minimizing the distances between the representations of corresponding word pairs (Zou et al., 2013). In general, it is not possible to align word vectors perfectly between languages because all of the texts the model is trained on and all of the semantic relations have a cultural background. It is even possible to observe cultural processes in texts from different points in times (Kozlowski et al., 2019). This also applies to our problem, e.g., while it is common in Germany to have savoury dishes for breakfast, in France sweet breakfasts are strongly preferred. The fact that word embeddings represent cultural characteristics has the consequence that they can also amplify biases, e.g. gender stereotypes (Bolukbasi et al., 2016). Fortunately, for our analysis this is of minor importance as we deal with food product data, which does not seem to be affected by such biases.

Fine-tuning pretrained embeddings. We use aligned Multilingual Unsupervised and Supervised Embeddings (MUSE) by Joulin et al. (2018), who make use of a Generative Adversarial Network and Procrustes for fine-tuning. They outperform other embedding models on many benchmark tasks (Joulin et al., 2018). When using the embeddings for training we make use of the existence of shared word forms in the vocabularies like *pizza*. In fact, 13% of the German word forms covered by the MUSE embeddings also appears in the French MUSE embeddings' vocabulary. For these shared words we calculate the average embedding vector and afterwards we combine both embedding spaces into one with the effect that the shared vocabulary will already be fine-tuned to the task, and this fine-tuning is directly transferred as the French model uses the same embedding space.
The pre-trained German embeddings cover only 47% and the French 75% of unique tokens in data. Subword embeddings are not provided with the published MUSE models, and, at least for transfer learning, aligning sub-words is a topic of its own since they carry much more syntactic then semantic information (Kayi et al., 2020).

Expanding Vocabularies using K-Nearest-Neighbour (KNN). We generate embeddings for words not covered by the pre-trained MUSE embedding vocabularies based on their distribution in our data set using a KNN approach (cf. Algorithm 1). Specifically, we trained a fastText model using the Gensim implementation (Řehůřek and Sojka, 2010) on 500,000 German and 330,000 French web scraped products.

While this amount of data is not sufficient to learn good semantics, the model still learns statistics which is especially important for the morphologically rich German language. It is an open research problem on how many web scraped products we would need to benefit significantly more, but our results, as we will see later, are still very good with the given data set.

From our own and the pre-trained fastText models we build a list of words that appeared in both models. This shared vocabulary is, in turn, used to generate MUSE-aligned embeddings for words in the data set that are not in the MUSE vocabulary. The aligned word vectors are calculated as a distance-weighted average of the MUSE embeddings of the shared vocabulary words that are nearest in the embedding space of our trained fastText model.

Algorithm 1 KNN-WEIGHTED-AVG-VECTORS(P, L, K)

Require: P, a dictionary with pre-trained word embeddings (keys are words, values the corresponding vectors); L, a dictionary with embeddings trained locally on the data set; K, number of nearest neighbours to consider

$SharedVocab \leftarrow P.keys \cap L.keys$
$OutOfVocab \leftarrow L.keys \setminus SharedVocab$
for all o in $OutOfVocab$ **do**
    $SD \leftarrow [\ ]$
    for all s in $SharedVocab$ **do**
        $SD \leftarrow SD \oplus \langle \text{DISTANCE}(L[o], L[s]), s \rangle$
    end for
    $SD \leftarrow \text{SORT}(SD)$ ▷ sort increasing by distance
    $\mathbf{v} \leftarrow \mathbf{0}$
    for $i = 1$ to K **do**
        $d, s \leftarrow SD_i$
        $\mathbf{p} \leftarrow P[s]$
        $\mathbf{v} \leftarrow \mathbf{v} + d\mathbf{p}$
    end for
    $\mathbf{v} \leftarrow \text{NORMALIZE}(\mathbf{v})$ ▷ e.g., to have 1.0 L2 norm
    $P[o] \leftarrow \mathbf{v}$ ▷ extend the pre-trained embeddings
end for

4.3 Network Architecture

We use MUSE embeddings extended by the method outlined in the previous subsection to represent the product texts. The classifier model itself is based on the aforementioned fastText framework. Since prediction of ECOICOP categories on the 5-digit level has never been done before for web scraped supermarket products, we investigate how well the classes can be predicted and how much labeled data is necessary in order to achieve satisfactory results. We differentiate between single language models, single language models for transfer learning and multilingual models. For the classification of the web scraped text strings we trained and tested different neural net architectures using the enriched MUSE embeddings as base.

Preparation. For the single language case we fine-tune word vectors during model training to adjust to the problem. The input sequences have a length of 34 tokens, which is the 95% quantile for our texts. Shorter documents are padded with zero vectors. To address the class imbalance we apply class weights to our network which are calculated as

$$weight_{class} = N/(C \cdot N_{class}),$$

where N is the number of products, C the number of classes and N_{class} the number of products in the class in question. The calculated class weights are taken into account during the learning phase. We train our model stepwise on 250, 500, 2,000, 10,000, 15,000 (French), and 25,000 (German) data points to report the fit at different levels. When training multilingual models we take the whole data of the source language while adding stepwise data of the target language. We oversample the target language in addition to balance the data. We split our data into training, validation, and test data sets. The held out test set includes products which appear neither in the training nor in the validation set. Therefore, we also measure the effect of changing product ranges.

Language	Train	Validation	Test
German	23,597	3,933	3,933
French	17,124	2,854	2,854

Table 2: Number of data points available for training, validation and test

Convolutional Neural Network. We make use of the idea to treat our products, represented as a sequence of vectors, as an image with the size $sequence\_length \times embedding\_dimension$ and use convolutions to extract features. This method was first used by Kim (2014). Using this technique the training can be fully parallelized and has today replaced RNNs in many NLP tasks (Gehring et al., 2017). We use multiple convolutional filters which move through the input sequence with the kernel looking at n tokens ($n \in \{3, 4, 5\}$) and all embedding coordinates (with a kernel size of $n \times 300$) at a time as proposed by Kim (2014). The used CNN looks at context windows of up to five words (Wang et al., 2018) and recognizes short-range dependencies, does not learn long-range dependencies as RNNs can. This reflects the characteristics of our data well. The outputs of the convolutions are pooled using maximum pooling, and the resulting outputs are concatenated and flattened into a one-dimensional vector. To regularize the network we apply dropout to 50% of the connections before the final dense layer which outputs a probability distribution over the 75 categories using softmax activation (Wang et al., 2018).

Single language training. Results of single language training with max pooling and trainable (non-static) embeddings, and average pooling with static embeddings for German on the test data are presented in Table 3.

Similarly, results for single language training on the French test data are presented in Table 4.

N	250	500	2,000	10,000	25,000
CNN_{max}	59.9	78.29	92.0	96.0	97.4
CNN_{avg}	58.8	63.2	88.3	94.2	95.3

Table 3: Accuracy on test data - training the CNN model on N German products in percent

N	250	500	2,000	10,000	15,000
CNN_{max}	58.5	82.8	90.8	95.0	96.2
CNN_{avg}	54.4	71.5	89.5	94.0	94.6

Table 4: Accuracy on test data training the CNN model on N French products in percent

For the German data we classified 96.0% of the test products correctly, training on 10,000 data points with max pooling. Our best French model reached 95.0% accuracy, trained on 10,000 data points with max pooling. Overall, the training curve is steep and the model learns even on small data. In addition, the training process is rather stable.

Transfer learning. Having built the classifier on our German product data set we want to make use of the MUSE property that the embeddings for individual languages are cross-lingually aligned, and reuse the German model on our French data. In particular, we examine the prospects of directly transferring knowledge by predicting the category of French products without the model having seen French data before (zero shot learning) and we also experiment with fine-tuning the German model with a small amount of French product data (few shot learning). To make this work we have to make a few changes to our model. Most important is the freezing of our word vectors. While we had them fine-tuned during single language training, now we need to ensure that they stay aligned. To further train the the model on French data we cut the last layer of the German model and replace it with a new initialized layer. In Table 5 we report scores both using max and average pooling layers.

Comparing the results with single language training we observe that, especially for cases where there is little training data available (up to 1,000 examples), transfer learning brings significant advantage compared to training models only on the corresponding monolingual data for each language.

Multilingual training. Results also confirm a higher advantage when both languages are trained jointly (cf. Table 6). Initially, with few samples, this is much higher then later when more training data is available. However, even then a small advantage can be observed.

N	0	100	250	1,000	5,000
CNN_{max}	42.0	63.9	78.7	87.4	93.3
CNN_{avg}	38.0	60.0	77.8	88.7	93.7

Table 5: Accuracy on test data training the transferred CNN model on N French products in percent

N	250	500	2,000	10,000	25,000
German	65.8	78.6	90.7	95.9	97.8
French	74.6	82.2	91.6	96.0	98.2

Table 6: Accuracy on test data training the CNN model on N French and German products in percent

Summary. We present in Figure 3 a comparison of all training approaches we experimented with. The CNN SL learning curve corresponds to the approach of using single-language training. CNN ML describes multi-language training, and the remaining curves show the characteristics of transfer learning, with static (CNN SE) and non-static embeddings (CNN NSE). Transfer learning has been shown to only

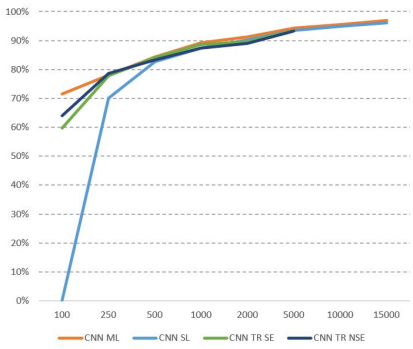

Figure 3: Accuracy on test data training the CNN models on N French products, single language training (CNN SL), multi language training (CNN ML), transfer learning with static embedding (CNN SE) and non-static (CNN NSE), in percent

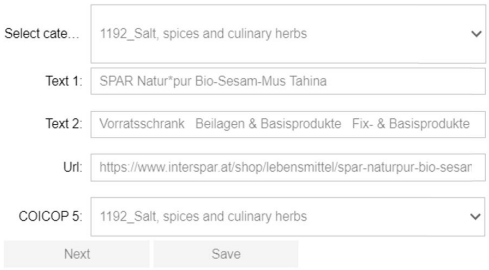

Figure 4: Labeling tool with a drop-down list to select the category one wants to label, information about the product and drop-down to select the correct label, pre-filled with the model's prediction

partially transfer results from German to French. Transferred models cannot be used for automation without additional supervised training, but can still provide predictions useful in certain settings, e.g. as annotation aids, as will be shown in the next section. In contrast, the joint German and French training achieved very good results and outperformed French single language training on every amount of data as well as the results from transfer learning. Especially when trained on all data available the multi language training raised the result by 2%. We therefore see an improvement for the low resource language when trained jointly.

5 Active Learning

Motivation. The main idea of active learning is to achieve higher model performance on the same amount of data or the same performance on less data by letting the learning algorithm choose which data points to label and train on. Using this approach we can take advantage of our transferred results to reduce annotation costs when transferring to other European and non-European languages.

Active learning is an active research field and strategies often differ for certain use cases. Methods suitable for our problem belong to the field of pool-based sampling (Settles, 2009) as we have a large pool of web-scraped products and it has to be decided which products should be labeled. One popular sampling strategy is uncertainty sampling: this approach always labels and learns from the data point the model is most uncertain about, with the goal to concentrate on the observation from which the model can acquire the maximal amount of new information:

$$x_{least\ confident} =_x P(\hat{y}|x).$$

This strategy is often refined by taking the distance to the second most likely class as measurement. This variant is called margin sampling (Settles, 2009). The exact opposite is suitable for our zero shot

prediction. As we start without annotated data, every observation in the new language is valuable in the beginning and we use the ones we are most certain about. To sample from different ECOICOP classes we condition the sampling on highest prediction certainty for a specific class.

Enhancing the annotation process for new languages. In this context, our transfer learning results give valuable input. The data creation process means adapting the classifier to a new language without having any labeled data. While uncertainty sampling is often used to find training examples where the model learns most, at the beginning of the annotation process every observation is valuable. Hence, we propose the following approach:

1. At the beginning we use zero shot predictions from the existing (transferred) model to create label proposals and use certainty sampling, i.e. select the ones the model is most certain about and present it to the human annotator for confirmation. This saves significantly time for annotating the first examples.

2. As we have shown above, the learning curve is very steep. Already 250 annotated observations increased the accuracy to 80%, which is double the rate of zero shot learning. After retraining the model with the 250 annotated examples, we switch to conditional certainty sampling and label further data points chosen both on the basis of the model's certainty and with the aim of generating an equally distributed data set which is beneficial for training.

3. In the last step, we propose to select the data points from which the classifier can learn the most, i.e. where it is most uncertain, using uncertainty sampling.

Labeling tool. We present in Figure 4 a screenshot of our labeling tool that implements the aforementioned process. Based on a given data set of unlabeled products, it tries to predict the correct labeling according to ECOICOP (5 digits) and presents this to the user. The user can simply look if it makes sense for the description, and when in doubt they can also open the product URL. If the label is correct then the human annotator simply clicks save and can continue with the next product. Otherwise the annotator needs to select the correct category. Since ECOICOP is hierarchical the annotator might simply select the correct label from another nearby category at a higher level, i.e., a label close to the predicted one.

Preliminary results. The benefits of the labeling tool were measured in a preliminary evaluation setup by two human annotators for French product data. Their feedback indicated that by using the proposals based on zero-shot learning they could accelerate the annotation process by a magnitude of 2. After retraining the classifier with the newly annotated data (ca. 1,000 labels) they could increase annotation speed by a magnitude of 3. This means just 20-30% of the time is needed compared to random annotation. These results are very encouraging, although a more thorough investigation is required with more languages as well as non-food ECOICOP categories.

6 Conclusion and Further Research

We investigated in this work how web scraped product data ca be classified in to the ECOICOP classification schema and how multilingual transfer learning can be applied to transfer results between languages. We demonstrated that zero-shot learning can be very useful, especially in the early phases when no manually labeled data is available. In the future we want to extend this work to further European languages and explore if some language combinations benefit more than others from the transfer from a pretrained single-language classification model and if a multilingual model generalizes for transfer learning. We found the predicted scores for the categories to be a good proxy for the certainty of the model and we expect that the generic framework introduced here can be reused for many different languages by changing only language-specific parts. This would help to focus the manual classification effort for millions of products that needs to be enhanced on a continuous basis. Furthermore, we want to include all ECOICOP categories in a generalized manner.

References

Jacek Białek and Maciej Berkesewicz. 2020. Scanner data in inflation measurement: from raw data to price indices. *arXiv: Applications*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357.

Alberto Cavallo and Roberto Rigobon. 2016. The billion prices project: Using online prices for measurement and research. *Journal of Economic Perspectives*, 30(2):151–178.

Alberto Cavallo, Gita Gopinath, Brent Neiman, and Jenny Tang. 2019. Tariff passthrough at the border and at the store: evidence from us trade policy. Technical report, National Bureau of Economic Research.

Jindong Chen, Yizhou Hu, Jingping Liu, Yanghua Xiao, and Haiyun Jiang. 2019. Deep short text classification with knowledge powered attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6252–6259.

Akiko Eriguchi, Melvin Johnson, Orhan Firat, Hideto Kazawa, and Wolfgang Macherey. 2018. Zero-shot cross-lingual classification using multilingual neural machine translation. *arXiv preprint arXiv:1809.04686*.

European Union. 2016. Regulation (eu) 2016/792 of the european parliament and of the council of 11 may 2016 on harmonised indices of consumer prices and the house price index, and repealing council regulation (ec) no 2494/95. *Official Journal of the European Union*, 59(24 May).

Eurostat. 2020. Faq - eurostat. `https://ec.europa.eu/eurostat/web/hicp/faq`. (Accessed on 06/12/2020).

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80, 12.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. *arXiv preprint arXiv:1804.07745*.

Efsun Sarioglu Kayi, Vishal Anand, and Smaranda Muresan. 2020. Multiseg: Parallel data and subword information for learning bilingual embeddings in low resource scenarios. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 97–105.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India, December. The COLING 2012 Organizing Committee.

Timo Koskimäki and Mari Ylä-Jarkko. 2003. Segmented markets and cpi elementary classifications. In *seventh meeting of the International working group on price indices, Paris*, pages 27–29.

Austin C Kozlowski, Matt Taddy, and James A Evans. 2019. The geometry of culture: Analyzing the meanings of class through word embeddings. *American Sociological Review*, 84(5):905–949.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827*.

Paweł Macias and Damian Stelmasiak. 2019. Food inflation nowcasting with web scraped data. Technical report.

Hazel Martindale, Edward Rowland, and Tanya Flower. 2019. Semi-supervised machine learning with word embedding for classification in price statistics. In *16th Meeting of the International Working Group on Price Indices*, Rio de Janeiro, Brazil.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Ben Powell, Guy Nason, Duncan Elliott, Matthew Mayhew, Jennifer Davies, and Joe Winton. 2018. Tracking and modelling prices using web-scraped price microdata: towards automated daily consumer price index forecasting. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 181(3):737–756.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA.

Tulika Saha, Sriparna Saha, and Pushpak Bhattacharyya. 2019. Tweet act classification: A deep learning based classifier for recognizing speech acts in twitter. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Seungwan Seo, Czangyeob Kim, Haedong Kim, Kyounghyun Mo, and Pilsung Kang. 2020. Comparative study of deep learning-based sentiment classification. *IEEE Access*, 8:6861–6875.

Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. 2017. Combining knowledge with deep convolutional neural networks for short text classification. In *IJCAI*, volume 350.

Shiyao Wang, Minlie Huang, and Zhidong Deng. 2018. Densely connected cnn with multi-scale feature attention for text classification. In *IJCAI*, pages 4468–4474.

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398.