

# Company Industry Classification with Neural and Attention-Based Learning Models

Stanislav Slavov  
Department of Mathematics,  
Princeton University,  
Princeton, NJ, United States  
sslavov@princeton.edu

Nikola Tulechki  
Sirma AI trading as Ontotext  
Sofia, Bulgaria  
nikola.tulechki@ontotext.com

Andrey Tagarev  
Sirma AI trading as Ontotext  
Sofia, Bulgaria  
andrey.tagarev@ontotext.com

Svetla Boytcheva  
Sirma AI trading as Ontotext &  
Institute of Information and Communication Technologies,  
Bulgarian Academy of Sciences,  
Sofia, Bulgaria  
svetla.boytcheva@ontotext.com

**Abstract**— This paper compares different solutions for the task of classifying companies with an industry classification scheme. Recent advances in deep learning methods show better performance in the text classification task. The dataset consists of short textual descriptions of companies and their economic activities. Target classification schemes are built by mapping related open data in a semi-controlled manner. Target classes are built from the bottom up by DBpedia. For the experiments are used modifications of methods BERT, XLNet, Glove and ULMfit with pre-trained models for English. Two simple models with perceptron architecture are used as the baseline. The results show that the best performance for multi-label classification of DBpedia companies abstracts is achieved by BERT and XLnet models, even for unbalanced classes.

**Keywords**— *Text-based Classification, Deep Learning, Big Data Applications, Multi-label Classification, Open Data*

## I. INTRODUCTION

With big data, the task of integrating and collating information from different sources is very important. The challenges of this task are represented by the different representations of data in the datasets and the possible inconsistency of the information. In addition, the same concepts are often presented in different ways. Different sources follow different ontologies that could treat the same concept differently. Thus, the task of achieving a coherent classification concept becomes important.

The task of classifying a company according to different industry classifications is challenging even for a person. Automating this process requires that two classification schemes can be mapped against each other. This ontology mapping is not always straightforward and often lacks full consistency.

We examine the task of company industry classification and present different approaches to the problem. Consistent classification of companies might be important for classifying a new company's equity on the market or for the consistency of macroeconomic data aggregated across industries. We approach the problem as a multi-label

classification, where the input is a company description from DBpedia<sup>1</sup>, and the output is a list of industries from a defined set  $L = \{l_1, l_2, \dots, l_n\}$ .

There are numerous company industry classifications of varying granularity. Some of the most popular are: Global Industry Classification Standard (GICS)<sup>2</sup>, Thomson Reuters Business Classification (TRBC)<sup>3</sup>, Industry Classification Benchmark (ICB)<sup>4</sup>, and International Standard Industrial Classification of All Economic Activities (ISIC)<sup>5</sup>.

## II. TEXT-CLASSIFICATION METHODS

Vast amount of information is still available in an unstructured format and requires the development of advanced techniques for structuring it.

There are no benchmarks datasets for industry classification task and the comparison of methods from the literature review is not feasible for this task. In the mid 90's there were performed many experiments [1–4] with "Industry Sector" data, that contain 6K company descriptions from the Web, classified in 70 industry sectors. However, the recent advances in text-based classification methods and the size of our datasets can consider such results as a little bit outdated.

The comparison [5] of the recently used methods for text-based classification shows that the classical methods like Support Vector Machine, Naïve Bayes (NB), Random Forest (RF) predominate the usage of other solutions.

The application of methods over "Industry Sector" data show for NB [1] accuracy up to 0.74, multinomial NB that deal with unbalanced classes [2] achieves significant improvement in the performance, error-correcting codes method [3] — accuracy up to 0.886, for Maximum Entropy

<sup>1</sup> <https://wiki.dbpedia.org>

<sup>2</sup> <https://www.msci.com/gics>

<sup>3</sup> <https://www.refinitiv.com/en/financial-data/indices/trbc-business-classification>

<sup>4</sup> <https://www.ftserussell.com/data/industry-classification-benchmark-icb>

<sup>5</sup> [https://unstats.un.org/unsd/publication/seriesM/seriesm\\_4rev4e.pdf](https://unstats.un.org/unsd/publication/seriesM/seriesm_4rev4e.pdf)

The research presented in this paper is partially funded by the project "The Intelligent Matching and Linking of Company Data" (CIMA), grant BG16RFOP002-1.005-0168-C01 by the EU's ERDF, OP „Innovations and Competitiveness 2014-2020" call "Intelligent Specialization".

classifier [4] reaches accuracy 0.788. In [6] is proposed SVM method based on one-vs-all and error-correcting output coding that outperforms NB accuracy for the "Industry Sector" dataset.

Because our dataset is based on DBpedia, the performance of neural networks (NN) algorithms over it for text-based classification task is of primary interest for us.

Glove [8] is a word representations model, based on so-called Global vectors — a new global log-bilinear regression model. In this method, the learning is based on non-zero elements in the word-word co-occurrence matrix.

In ELMo for each token is assigned a representation which is a function for the whole input sentence. XLNet incorporates ideas from Transformer-XL [11] and achieves the best performance for DBpedia with the minimal error 0.62, where "*classification error*" is defined as 1.0 minus classification accuracy. BERT [13] performance demonstrates error 1.09 for the same dataset, while the BERT<sub>Large</sub> error for the DBpedia is 0.64.

These results motivate us to examine the performance of XLnet, BERT, ULMfit, and Glove for the company industry classification tasks.



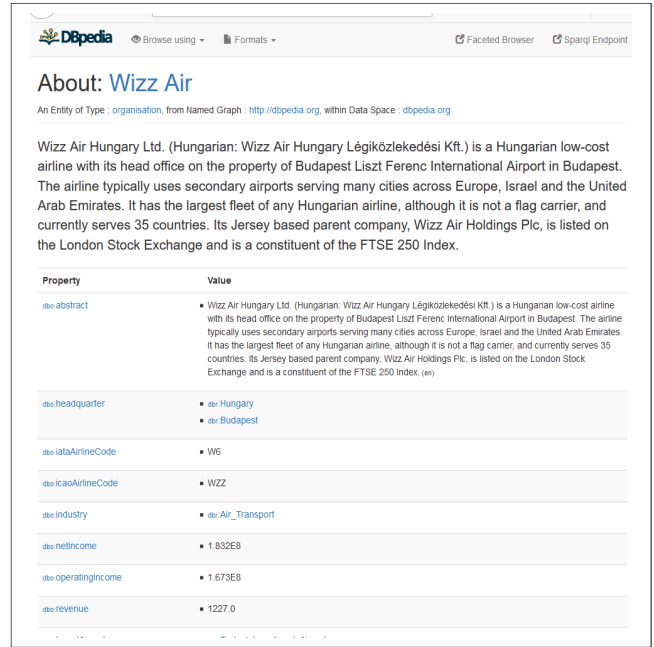
### III. DATASET

Our experiments were carried out on an extension of the English DBpedia Knowledge Base with some custom remapping of properties to create a unified hierarchical classification of companies. Of particular note are the approximately 300 thousand organizations which have articles in the English Wikipedia as we will use their abstracts (i.e. descriptions of up to 100 words) as the textual input for the classification algorithms (see Fig. 2).

The taxonomy was extracted from 1289 tags, articles and classes that are frequently used in Wikipedia to classify organizations. From that unorganized set of labels, we built a new taxonomy consisting of 612 classes arranged in a hierarchy with 32 top-level industries and up to six layers of depth (e.g. "Entertainment and publishing" → "Media and publishing" → "Mass Media" → "Broadcasting" → "Radio" → "Internet Radio"). Fig. 1 shows the Industry hierarchy for the "Transport" top-level class. For this particular class hierarchy consists of three levels, the first level containing subclasses such as "Air Transport" and "Logistics" and a second even more specific level with industries such as "Airline" and "Aircraft Maintenance". Also, are shown (in yellow) equivalent classes for some elements of the hierarchy. "Ship Transport", for example, is a synonym for "Maritime Transport". The nuances of the complete hierarchy and a small number of organizations in the lower levels were too complex for the needs of our experiment, so the actual experiments were only carried out on the 32 top-level classes. In the description of Wizz Air airline in DBpedia (see Fig. 2) is used as industry tag "Air\_Transportation", which is a subindustry of the top-level sector "Transport". An organization's description in DBpedia can contain multiple values for the industry. Fig. 3 shows that Wizz Air airline has industry relations "Air\_Transport", "Aerospace", "Airline" and "Aviation", but it is not related to the top-industry "Transport".

The method of building up this taxonomy from the existing data means that we are working on an unbalanced sparse multi-class multi-label classification task. The task is multi-class by definition since we have chosen to work with the 32-top classes in taxonomy. It is multi-label because there is no inherent exclusivity to these classes in the underlying data and in fact, 56.5% of organizations in our dataset have multiple labels. It is sparse because the taxonomy is not exhaustive and labels are not reliably applied when appropriate which accounts for the 27% organizations in our dataset with no assigned labels. Finally, it is unbalanced because, as we will discuss later in more detail, individual top-level classes contain anywhere between a few hundred to several tens of thousands of organizations.

The full dataset used in these experiments is publicly available as part of the FactForge<sup>6</sup> demonstrator.



Property	Value
<code>dbp:abstract</code>	Wizz Air Hungary Ltd. (Hungarian: Wizz Air Hungary Légitársaság Kft.) is a Hungarian low-cost airline with its head office on the property of Budapest Liszt Ferenc International Airport in Budapest. The airline typically uses secondary airports serving many cities across Europe, Israel and the United Arab Emirates. It has the largest fleet of any Hungarian airline, although it is not a flag carrier, and currently serves 35 countries. Its Jersey based parent company, Wizz Air Holdings Plc, is listed on the London Stock Exchange and is a constituent of the FTSE 250 Index. (en)
<code>dbp:headquarter</code>	<ul style="list-style-type: none"> <li><code>dbp:Hungary</code></li> <li><code>dbp:Budapest</code></li> </ul>
<code>dbp:ianaAirlineCode</code>	W6
<code>dbp:icaoAirlineCode</code>	WZZ
<code>dbp:industry</code>	<code>dbp:Air_Transport</code>
<code>dbp:netIncome</code>	1.832E8
<code>dbp:operatingIncome</code>	1.673E8
<code>dbp:revenue</code>	1227.0

Fig. 2. Wizz Air description in DBpedia.

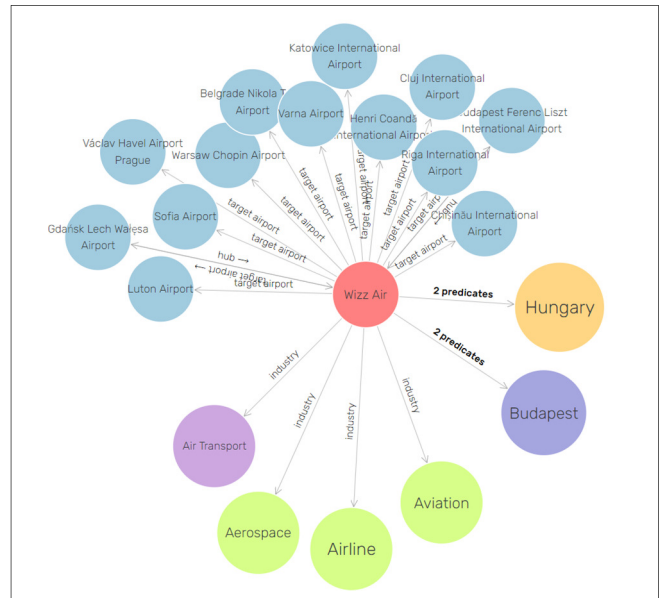


Fig. 3. Wizz Air relations in FactForge.

### IV. EXPERIMENTS

Recent advances in deep learning and neural network models show better performance in the text-based classification tasks. We use as baseline two simple methods with perceptron architecture.

For the solution of the company industry classification task are compared four more advanced methods BERT, XLnet, Glove, and ULMfit. All these methods are characterized by high-time and memory complexity. Our experiments use pre-trained vectors for the English language.

<sup>6</sup> <http://factforge.net>

### A. Linear Baseline — Hot-unigram

A perceptron with no hidden layers serves as a baseline for the experiments. The company descriptions are first processed through a standard NLP pipeline for stopword removal and stemming and then each unigram is represented as a one-hot vector representation<sup>7</sup>{binary vectors that are all zero values except for the index corresponding to the word or n-gram}. The perceptron processes as a sequence bit taking as input the sum of the vectors corresponding to the unigrams in the sequence.

### B. Hot-bigram

The second approach is a customized linear model. It is identical to the baseline model, except that features include unigrams and bigrams.

### C. GloVe

The third approach is the first among the methods that utilize the transfer learning framework for downstream tasks such as ours. It uses the same preprocessing steps as the linear baseline approach (i.e. NLP pipeline for stopword removal and stemming, resulting text is processed into unigrams) but instead of one-hot vectors, the pre-trained GloVe vector embeddings are used. GloVe is a model of 300-dimensional word embeddings trained on the large Common Crawl corpus of 840 billion tokens with a vocabulary of 2.2 million words. While there is no additional training of the GloVe embeddings for our specific data, our text data is significantly smaller in size compared to the corpus used for pre-training.

The resulting vectors are once again passed to a linear perceptron. Training times were significantly lower (under a minute compared to 15-60 minutes) because the 300-dimensional resulting vector is usually much smaller than the previous one-hot representation.

### D. ULMfit

The first state-of-the-art approach we tested is ULMfit [12] by fast.ai<sup>7</sup>. The first thing that distinguishes it from the baseline algorithms is the text preprocessing- rather than the aforementioned NLP stemming pipeline feeding into one-hot vectors, this approach uses all available text abstracts in order to train a fully custom Language Model based on AWD-LSTM which produces context vectors on the organization description level. For the purposes of these experiments, we used the default settings for the network due to time restrictions but there is an opportunity for further exploration of the language model's performance with varying initial parameters.

The classification training step is implemented as an additional layer added onto an already trained language model. The effect allows relative quick initial training of the context vectors followed by some fine-tuning of the context vectors along with the specific classification layer training. In our experiments, the classification layer only took a few epochs to converge to a stable solution in each case but the combined fine-tuning of classification model and language

model was much more computationally expensive and the results reported here did not include any fine-tuning.

### E. BERT

BERT is an autoencoder model using the Transformer architecture that takes in a word sequence and encodes it into a representation, randomly masking some of the tokens. The objective of the algorithm during training is to correctly predict, at the stage of decoding, the masked tokens given the rest of the sequence as context. Through this procedure, BERT implements a bidirectional context, an advantage over autoregressive language models that can only use prior content as context. We used the pre-trained word embeddings and the vocabulary given by the BERT-Base model, utilizing the transfer learning framework BERT provides for downstream tasks. The embeddings themselves have been trained from scratch on BooksCorpus (800M words) and English Wikipedia (2500M words). For multi-label classification, we added a final layer to the decoder taking the output and connecting to a layer of sigmoids, one for each class, where a sigmoid's activation is interpreted as the algorithm deciding to assign the respective label. For this purpose, the algorithm was adapted to read a label as a binary vector representing all the labels as ones in the respective positions.

The initial dataset was split at a ratio 4:1:1 of training, evaluation and test sets with proportional amounts of examples for each class (Class sizes shown in Table II are for the whole dataset). The data was pre-processed by the standard pipeline before tokenization and masking. The algorithm works with sequences of a prespecified length and will either pad or cull the sequence to match the exact size specified. We used a sequence length of 128 because most company descriptions are about 100 words long. In total, our training set contains about 15-20M words, which is substantially less than the pre-training corpus. We fine-tuned for 20 epochs on the training set with batch size 32 and learning rate 2e-5. With a configuration of 4 NVIDIA Tesla T4 GPUs of 16GB memory each, one fine-tuning epoch ran for about 80 minutes.

### F. XLNet

XLNet is an autoregressive model that uses TransformerXL — an improved version of the BERT architecture. The model uses the same transfer learning approach to provide fine-tuning for downstream tasks, where pretraining was done on a larger dataset of about 32.89B words. The key difference between XLNet and BERT is that XLNet is a permutation language model: as a sequence is processed, random permutations of it are examined and for each the model seeks to predict the tokens in that order, taking previous tokens as context. To replace the token masking present in BERT, XLNet introduces Two-Stream Attention. In BERT, the encoder codes positional and content information in the representation of a token. For the purposes of the permutation model, XLNet maintains two streams containing two representations: the content stream, containing both positional and content information up to the current token, and a query stream - containing only content information for previous tokens and only position information

<sup>7</sup> <https://github.com/jannenev/ulmfit-language-model>

for the current token to be predicted. This second stream mirrors the masking by hiding the current content and previous positions. This is one of the main advantages of XLNet over BERT: it is an autoregressive model that implements a bidirectional context and also keeps the Transformer architecture, enabling the same transfer learning

framework used in BERT.

The data is preprocessed by a standard pipeline before tokenization and masking. The same procedure of formatting sequences to a fixed length is applied and we used the same length of 128.

TABLE I. COMPARISON OF MICRO AND MACRO PRECISION, RECALL, AND F1 BETWEEN APPROACHES

	hot unigram			hot bigram			Glove			ULMfit			BERT			XLNet		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
<b>Micro</b>	0.943	0.901	0.922	0.944	0.909	0.926	0.913	0.899	0.906	<b>0.958</b>	0.898	0.927	0.942	<b>0.938</b>	0.940	0.945	<b>0.938</b>	<b>0.941</b>
<b>Macro</b>	0.788	0.625	0.689	0.786	0.658	0.712	0.700	0.674	0.686	<b>0.868</b>	0.600	0.671	0.805	0.766	0.783	0.811	<b>0.771</b>	<b>0.789</b>

TABLE II. CLASS-BY-CLASS COMPARISON OF F1 SCORES BETWEEN THE FOUR ALGORITHMS

Industry	Size	hot-unigram			hot-bigram			Glove			ULMfit			BERT			XLNet		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Entertainment and publishing	76309	0.98	0.97	<b>0.98</b>	0.98	0.98	<b>0.98</b>	0.97	0.96	0.96	0.99	0.97	<b>0.98</b>	0.98	0.98	<b>0.98</b>	0.98	0.99	<b>0.98</b>
Education	55221	0.98	0.97	0.98	0.98	0.98	0.98	0.98	0.96	0.97	0.99	0.99	<b>0.99</b>	0.99	0.99	<b>0.99</b>	0.99	0.99	<b>0.99</b>
Travel and sport	44768	0.99	0.98	<b>0.99</b>	0.71	0.65	0.68	0.91	0.89	0.90	0.99	0.97	0.98	0.99	0.99	<b>0.99</b>	0.99	0.98	<b>0.99</b>
Public sector	26391	0.97	0.96	0.97	0.72	0.56	0.63	0.98	0.98	<b>0.98</b>	0.98	0.97	<b>0.98</b>	0.98	0.98	<b>0.98</b>	0.98	0.99	<b>0.98</b>
Information technology	10255	0.87	0.77	0.82	0.97	0.97	<b>0.97</b>	0.70	0.65	0.67	0.81	0.80	0.81	0.83	0.85	0.84	0.82	0.87	0.84
Transport	10007	0.90	0.82	0.86	0.99	0.99	<b>0.99</b>	0.81	0.78	0.79	0.96	0.91	0.93	0.96	0.94	0.95	0.96	0.95	0.96
Manufacturing	7757	0.94	0.91	<b>0.93</b>	0.94	0.92	<b>0.93</b>	0.73	0.71	0.72	0.74	0.63	0.68	0.68	0.71	0.69	0.74	0.68	0.71
Financial services	6086	0.82	0.76	0.79	0.90	0.83	0.87	0.65	0.58	0.61	0.93	0.83	0.88	0.92	0.88	<b>0.90</b>	0.91	0.89	<b>0.90</b>
Retail	4464	0.83	0.63	0.72	0.87	0.80	<b>0.84</b>	0.63	0.69	0.66	0.86	0.59	0.70	0.81	0.76	0.78	0.83	0.72	0.77
Food and beverage	3748	0.76	0.55	0.64	0.87	0.79	0.83	0.61	0.56	0.59	0.82	0.85	0.84	0.87	0.91	<b>0.89</b>	0.89	0.90	<b>0.89</b>
Nonprofit organization	3655	0.71	0.64	0.67	0.78	0.62	0.69	0.72	0.65	0.68	0.90	0.80	0.85	0.88	0.89	<b>0.89</b>	0.89	0.86	0.87
Personal and household goods	3206	0.79	0.64	0.70	0.84	0.70	0.76	0.96	0.95	<b>0.96</b>	0.75	0.52	0.61	0.71	0.72	0.72	0.71	0.69	0.70
Automotive	2564	0.84	0.71	0.77	0.84	0.73	0.78	0.46	0.45	0.46	0.83	0.71	0.77	0.86	0.83	<b>0.84</b>	0.83	0.83	0.83
Telecommunications	2500	0.94	0.85	<b>0.89</b>	0.74	0.61	0.67	0.77	0.69	0.73	0.87	0.67	0.76	0.77	0.78	0.78	0.75	0.8	0.77
Aerospace and defense	2425	0.78	0.61	0.69	0.72	0.54	0.62	0.76	0.76	<b>0.76</b>	0.90	0.54	0.68	0.74	0.71	0.73	0.76	0.76	<b>0.76</b>
Engineering	1758	0.50	0.21	0.30	0.84	0.76	0.80	0.86	0.82	<b>0.84</b>	0.78	0.25	0.37	0.65	0.58	0.61	0.66	0.57	0.61
Utility	1599	0.61	0.37	0.46	0.64	0.43	0.52	0.87	0.84	<b>0.86</b>	0.83	0.59	0.69	0.78	0.77	0.78	0.76	0.78	0.77
Commercial & professional services	1268	0.72	0.52	<b>0.61</b>	0.62	0.41	0.49	0.54	0.52	0.53	0.71	0.06	0.11	0.61	0.59	0.60	0.70	0.46	0.55
Fossil fuel	1213	0.81	0.68	0.74	0.64	0.47	0.54	0.76	0.72	0.74	0.89	0.67	0.76	0.83	0.79	<b>0.81</b>	0.77	0.84	0.80
Cultural heritage	1139	0.79	0.61	0.69	0.81	0.71	0.76	0.49	0.48	0.48	0.93	0.89	0.91	0.92	0.92	0.92	0.94	0.94	<b>0.94</b>
Pharmaceuticals and life sciences	1062	0.86	0.77	<b>0.81</b>	0.77	0.65	0.71	0.75	0.71	0.72	0.83	0.71	0.77	0.84	0.79	<b>0.81</b>	0.86	0.76	<b>0.81</b>
Real estate	920	0.66	0.43	0.52	0.59	0.38	0.46	0.42	0.39	0.41	0.86	0.55	0.67	0.83	0.64	0.73	0.81	0.69	<b>0.74</b>
Healthcare	915	0.66	0.37	0.48	0.83	0.69	0.75	0.62	0.53	0.57	0.80	0.40	0.53	0.65	0.57	0.61	0.69	0.66	<b>0.68</b>
Marketing	902	0.68	0.33	0.44	0.68	0.43	0.53	0.54	0.54	0.54	0.84	0.31	0.46	0.74	0.79	0.76	0.78	0.79	<b>0.78</b>
Conglomerate (company)	780	0.85	0.64	0.73	0.82	0.68	0.74	0.80	0.81	<b>0.81</b>	0.83	0.16	0.27	0.61	0.42	0.50	0.56	0.43	0.49
Construction and materials	764	0.58	0.33	0.42	0.79	0.65	<b>0.72</b>	0.51	0.49	0.50	0.64	0.19	0.29	0.61	0.57	0.59	0.71	0.60	0.65
Mining	665	0.83	0.64	0.72	0.93	0.89	<b>0.91</b>	0.70	0.64	0.66	0.88	0.61	0.72	0.88	0.67	0.76	0.90	0.71	0.80
Justice and law	577	0.73	0.43	0.54	0.94	0.86	0.90	0.73	0.71	0.72	0.92	0.92	0.92	0.97	0.95	0.96	0.98	0.96	<b>0.97</b>
Chemical industry	526	0.74	0.45	0.56	0.63	0.38	0.47	0.49	0.48	0.48	0.79	0.09	0.15	0.73	0.65	<b>0.69</b>	0.71	0.64	0.67
Agriculture	359	0.58	0.22	0.32	0.70	0.43	0.53	0.38	0.40	0.39	0.66	0.05	0.10	0.70	0.54	0.61	0.73	0.66	<b>0.69</b>
Forest and paper	328	0.91	0.85	<b>0.88</b>	0.47	0.30	0.36	0.41	0.37	0.39	0.71	0.20	0.31	0.70	0.70	0.70	0.67	0.75	0.71
Metal	302	0.61	0.35	0.45	0.59	0.27	0.37	0.89	0.88	<b>0.88</b>	0.69	0.23	0.34	0.74	0.63	0.68	0.71	0.56	0.62

## V. RESULTS AND DISCUSSION

We will begin the analysis of our results by comparing the averaged measures for recall, precision and F1 score presented in Table I. Micro scores are calculated over each individual example while macro scores are averaged over the respective scores for each class. We can see that all algorithms perform very well on the micro scores with the state-of-the-art algorithms having a small but significant advantage. Specifically, ULMfit gives us the best precision while XLnet achieves a much better recall and overall F1 score. Meanwhile, the macro scores are significantly worse all around but we still have ULMfit and XLnet as the champions in their respective categories, this time by an even wider margin.

A few important conclusions can be reached from observing this data. Firstly, the dataset is simple enough that even naive baseline approaches can achieve rather good results while their training times are much lower. By comparing their results on this simpler task, we intend to establish that the deep learning approaches match the performance of the baseline models on top-level industries while giving an opportunity to solve tasks that are beyond the capabilities of the baseline models entirely e.g. full hierarchical classification or company similarity.

Secondly, the state-of-the-art approaches do outperform the naive approaches but there is no unambiguous winner among them- there is a trade-off in precision versus recall between ULMfit and XLnet while BERT performs almost as well as XLnet. Still, XLnet has a solid advantage of F1 score over ULMfit so it performs the best of the tested algorithms on this task.

Finally, the large difference between micro and macro scores points to much worse overall performance on the less represented classes. For a more in-depth look at this disparity, let's examine the class-by-class results from Table II.

The second column shows us the size of each respective top-level industry in terms of the number of organizations assigned that label. We can immediately note that the largest industry has over 76 thousand examples while the smallest one has barely 300 and more than ten industries have fewer than a thousand examples. This does mean that the disparity in frequency between the largest and smallest classes is quite significant. It is also worth noting that all algorithms do fairly well on most big classes with all state-of-the-art approaches achieving very close results. However, once we get to classes with only a few thousand examples (meaning 99% negative examples in our dataset), the behavior of all algorithms becomes rather erratic.

It is evident that no single algorithm is the best approach for all smaller classes as each approach gets the best result in at least a few cases, however, XLnet and BERT seem to achieve more stable performance overall as indicated by their macro F1 scores due to having much better recall overall. The expectation from errors in the underlying data would be that smaller classes are less likely to be applied by the editors in Wikipedia, it seems likely that some of the lowered

precision is due to false "false positives" i.e. organizations that should be classified in a certain industry but are not in the data. If that is indeed the case, the performance of XLnet and BERT will likely be even better than the numbers suggest.

TABLE III. NUMBER OF TRAINING EPOCHS FOR STATE-OF-THE-ART ALGORITHMS

	ULMfit	BERT	XLNet
<i>Epochs</i>	16*	20	25

One final aspect worth examining is the training time spent on state-of-the-art algorithms. Table III shows the number of training epochs spent on training each algorithm but it is worth noting that ULMfit was trained on a less powerful machine so it not only had fewer training epochs but its language model was trained independently of the classifier network and no fine-tuning was performed. It is possible that with more training time and several epochs of fine-tuning, ULMfit could close the gap in performance to XLNet and it certainly already has an advantage in precision which is important for certain applications.

## VI. CONCLUSION AND FURTHER WORK

Overall, most algorithms achieve very good results on the larger classes and some of the state-of-the-art approaches even get very promising results on average on the smaller classes. From our experiments, XLnet achieved the highest scores slightly outperforming BERT in most cases and despite achieving a lower precision than ULMfit overall, it had a much better recall. While all three algorithms perform better than the baseline, XLnet provides shows the most promise on this task.

The next step would be to examine techniques for balancing the smaller classes (i.e. use some forms of undersampling or oversampling) and compare their effects on the performance of the various algorithms. Since all algorithms suffer to some degree in performance on the smaller classes, this has the potential to increase performance all around.

Another aspect to examine is the number of training epochs that each algorithm underwent. As we discussed, XLNet benefits from the largest number of epochs and longer training times might allow BERT and ULMfit to improve their performance. Especially in the case of ULMfit where no fine-tuning of the language model was attempted, the effect on performance could be significant.

A final and somewhat trivial but very important avenue of exploration would be to carry out some manual evaluation of the errors. As described, the crowd-sourced nature of Wikipedia means that the consistency in label application and use is less than uniform so it is reasonable to expect some of the "mistakes" made by the algorithms are actually errors in the underlying dataset. This would be especially true for any false-positive results in smaller classes where the human curators of Wikipedia were likely unfamiliar with the existence of the class at all and thus were unlikely to apply it.

#### ACKNOWLEDGMENT

The research presented in this paper is partially funded by the project “The Intelligent Matching and Linking of Company Data” (CIMA), grant BG16RFOP002-1.005-0168-C01 by the EU’s ERDF, OP „Innovations and Competitiveness 2014-2020” Call “Intelligent Specialization”.

#### REFERENCES

- [1] A. McCallum, and K. Nigam. "A comparison of event models for naive Bayes text classification." *AAAI-98 Workshop on learning for text categorization*. Vol. 752. No. 1. pp. 41-48, 1998.
- [2] E. Frank, and R. R. Bouckaert. "Naive Bayes for text classification with unbalanced classes." *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, Berlin, Heidelberg, pp. 503-510, 2006.
- [3] R. Ghani. "Using error-correcting codes for text classification." *ICML*, pp. 303-310, 2000.
- [4] K. Nigam, J. Lafferty, and A. McCallum. "Using maximum entropy for text classification." *IJCAI-99 workshop on machine learning for information filtering*. Vol. 1. No. 1, pp. 61-67, 1999.
- [5] J. Hartmann, et al. "Comparing automated text classification methods." *International Journal of Research in Marketing* Vol. 36. No. 1, pp. 20-38, 2019.
- [6] J. D. Rennie, and R. Ryan. "Improving multiclass text classification with the support vector machine." 2001.
- [7] S. Tan. "An effective refinement strategy for KNN text classifier." *Expert Systems with Applications* Vol. 30 No. 2, pp. 290-298, 2006.
- [8] J. Pennington, R. Socher, and C. Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532-1543, 2014.
- [9] M. E. Peters, et al. "Deep contextualized word representations." *arXiv preprint arXiv:1802.05365*, 2018.
- [10] Z. Yang, et al. "XLNet: Generalized Autoregressive Pretraining for Language Understanding." *arXiv preprint arXiv:1906.08237*. 2019.
- [11] Z. Dai, et al. "Transformer-xl: Attentive language models beyond a fixed-length context." *arXiv preprint arXiv:1901.02860*. 2019.
- [12] J. Howard, and S. Ruder. "Universal language model fine-tuning for text classification." *arXiv preprint arXiv:1801.06146*. 2018.
- [13] J. Devlin, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805*. 2018.