DOMAIN ADAPTATION IN NATURAL LANGUAGE PROCESSING

Inaugural-Dissertation
an der Ludwig-Maximilians-Universität München
zur Erlangung des Doktorgrades der Philosophie

vorgelegt von
Marina Sedinkina
aus Emangelinsk, Russische Föderation

München 2021

Erstgutachter: Prof. Dr. Hinrich Schütze
1.Korreferent: Prof. Dr. Klaus Schulz
2.Korreferent: Prof. Dr. Oliver Schallert

TAG DER MÜNDLICHEN PRÜFUNG: 09.03.2021

ABSTRACT

Domain adaptation has received much attention in the past decade. It has been shown that domain knowledge is paramount for building successful Natural Language Processing (NLP) applications.

To investigate the domain adaptation problem, we conduct several experiments from different perspectives. First, we automatically adapt sentiment dictionaries for predicting the financial outcomes "excess return" and "volatility". In these experiments, we compare manual adaptation of the domain-general dictionary with automatic adaptation, and manual adaptation with a combination consisting of first manual, then automatic adaptation. We demonstrate that automatic adaptation performs better than manual adaptation, namely the automatically adapted sentiment dictionary outperforms the previous state of the art in predicting excess return and volatility. Furthermore, we perform qualitative and quantitative analyses finding that annotation based on an expert's a priori belief about a word's meaning is error-prone – the meaning of a word can only be recognized in the context that it appears in.

Second, we develop the *temporal transfer learning* approach to account for the language change in social media. The language of social media is changing rapidly – new words appear in the vocabulary, and new trends are constantly emerging. Temporal transfer-learning allows us to model these temporal dynamics in the document collection. We show that this method significantly improves the prediction of movie sales from discussions on social media forums. In particular, we illustrate the success of parameter transfer, the importance of textual information for financial prediction, and show that temporal transfer learning can capture temporal trends in the data by focusing on those features that are relevant in a particular time step, i.e., we obtain more robust models preventing overfitting.

Third, we compare the performance of various domain adaptation models in low-resource settings, i.e., when there is a lack of large amounts of high-quality training data. This is an important issue in computational linguistics since the success of NLP applications primarily depends on the availability of training data. In real-world scenarios, the data is often too restricted and specialized. In our experiments, we evaluate different domain adaptation methods under these assumptions and find the most appropriate techniques for such a low-data problem. Furthermore, we discuss the conditions under which one approach substantially outperforms the other.

Finally, we summarize our work on domain adaptation in NLP and discuss possible future work topics.

Die Domänenanpassung hat in den letzten zehn Jahren viel Aufmerksamkeit erhalten. Es hat sich gezeigt, dass das Domänenwissen für die Erstellung erfolgreicher NLP-Anwendungen (Natural Language Processing) von größter Bedeutung ist.

Um das Problem der Domänenanpassung zu untersuchen, führen wir mehrere Experimente aus verschiedenen Perspektiven durch. Erstens passen wir Sentimentlexika automatisch an, um die Überschussrendite und die Volatilität der Finanzergebnisse besser vorherzusagen. In diesen Experimenten vergleichen wir die manuelle Anpassung des allgemeinen Lexikons mit der automatischen Anpassung und die manuelle Anpassung mit einer Kombination aus erst manueller und dann automatischer Anpassung. Wir zeigen, dass die automatische Anpassung eine bessere Leistung erbringt als die manuelle Anpassung: das automatisch angepasste Sentimentlexikon übertrifft den bisherigen Stand der Technik bei der Vorhersage der Überschussrendite und der Volatilität. Darüber hinaus führen wir eine qualitative und quantitative Analyse durch und stellen fest, dass Annotationen, die auf der a priori Überzeugung eines Experten über die Bedeutung eines Wortes basieren, fehlerhaft sein können. Die Bedeutung eines Wortes kann nur in dem Kontext erkannt werden, in dem es erscheint.

Zweitens entwickeln wir den Ansatz, den wir *Temporal Transfer Learning* benennen, um den Sprachwechsel in sozialen Medien zu berücksichtigen. Die Sprache der sozialen Medien ändert sich rasant – neue Wörter erscheinen im Vokabular und es entstehen ständig neue Trends. Temporal Transfer Learning ermöglicht es, diese zeitliche Dynamik in der Dokumentensammlung zu modellieren. Wir zeigen, dass diese Methode die Vorhersage von Filmverkäufen aus Diskussionen in Social-Media-Foren erheblich verbessert. In unseren Experimenten zeigen wir (i) den Erfolg der Parameterübertragung, (ii) die Bedeutung von Textinformationen für die finanzielle Vorhersage und (iii) dass Temporal Transfer Learning zeitliche Trends in den Daten erfassen kann, indem es sich auf die Merkmale konzentriert, die in einem bestimmten Zeitschritt relevant sind, d. h. wir erhalten robustere Modelle, die eine Überanpassung verhindern.

Drittens vergleichen wir die Leistung verschiedener Domänenanpassungsmodelle in ressourcenarmen Umgebungen, d. h. wenn große Mengen an hochwertigen Trainingsdaten fehlen. Das ist ein wichtiges Thema in der Computerlinguistik, da der Erfolg der NLP-Anwendungen stark von der Verfügbarkeit von Trainingsdaten abhängt. In realen Szenarien sind die Daten oft zu eingeschränkt und spezialisiert. In unseren Experimenten evaluieren wir verschiedene Domänenanpassungsmethoden unter diesen Annahmen und finden die am besten

geeigneten Techniken dafür. Darüber hinaus diskutieren wir die Bedingungen, unter denen ein Ansatz den anderen deutlich übertrifft.

Abschließend fassen wir unsere Arbeit zur Domänenanpassung in NLP zusammen und diskutieren mögliche zukünftige Arbeitsthemen.

PUBLICATIONS

Chapter 3 corresponds to the following publication:

Sedinkina, Marina, Nikolas Breitkopf, and Hinrich Schütze (July 2019). "Automatic Domain Adaptation Outperforms Manual Domain Adaptation for Predicting Financial Outcomes." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 346–359. DOI: 10.18653/v1/P19-1034. URL: https://www.aclweb.org/anthology/P19-1034.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my supervisor, Professor Dr. Hinrich Schütze, who I admire and respect for what he does. He enriched the research contained in this work by important insights, guided and encouraged me to pursue this thesis. I thank Hinrich Schütze for his time, the interest, the contemplation on issues this work seeks to address, and for the helpful comments on several draft versions of this work that not only improved it but also enriched my professional knowledge in Natural Language Processing. Thank you to be positive and patient even when sometimes the road got tough. Our meetings always inspired me to work harder and think outside of the box. Thank you for a comprehensive and objective critique. It was a great honour for me to work with you, one of the famous and greatest computational linguists. I am happy that I had a chance and an opportunity to become a part of the highly professional team at the Center for Information and Language Processing.

I wish to express my deepest appreciation to my second supervisor, Professor Dr. Klaus Schulz, who took the time to supervise my thesis and contribute to this work through his further comments.

I would like to pay my special regards to Dr. Nikolas Breitkopf, Assistant Professor of Institut for Finance and Banking. His important help and advice allowed me to make important contributions to this work.

Professor of the University of Vienna and the former colleague, Dr. Benjamin Roth, deserves a special mention for his involvement in my research. His support also allowed me to make important contributions to my research. Thank you for an endless stream of ideas and the invaluable assistance that you provided during my study.

I thank my dear colleagues who accompany me during my PhD, and who gave me very useful comments and advises concerning my projects and this thesis.

To conclude, I cannot forget to thank my family and friends for their support and great love. They kept me going on in these three very intense academic years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

This chapter reviews our motivation, defines the problems and presents the goals we want to achieve in order to investigate domain adaptation in Natural Language Processing.

1.1 MOTIVATION

The essence of a word meaning has always been one of the key questions in linguistics and philosophy of language. Recent findings regarding this issue have led to the conclusion that a word meaning arises from language use. Structural linguists reveal that a word meaning is established by the network of relations among a cluster of semantically related words (Gliozzo and Strapparava, 2009). Corpus studies confirm that the word meaning can only be activated by other words, i.e., in the context in which they are used (Hanks, 2000). In computational linguistics, the contexts are utilized to derive word representations – high dimensional vector space representations that encode the semantics of words (Sridharan and Murphy, 2012). The models used to obtain these representations are based on the idea of the distributional hypothesis (Harris, 1954), which suggests that the meaning of the word can be induced from a large number of texts. Hence, the opinion that a word derives its meaning from its context is widespread in linguistics.

Another issue concerning the question of word meaning is polysemy – the capacity of a word to have multiple meanings. Almost every word is more or less polysemous. Therefore, this requires a more careful interpretation of a word that involves the accommodation between what is given semantically, syntactically, but also what is inferred from the surrounding pragmatic context (Herman et al., 2003). In other words, the word meanings need to be understood against broader knowledge configurations, variously studied as "domains". Consider, as an illustration the word *God*, and the fact that it would be odd to say that this word is negative in some contexts. This word's main meaning is: "Almighty", creator of the universe. The problem arises not because the definition of *God* is wrong in any sense. Rather, it is because the concept of "God" needs to be understood in the context in which it occurs. For instance in the financial domain, the application of the word becomes problematic because this word implies a negative situation. In finance, this word is mostly used in the phrase "act of God", indicating that the company cannot fulfill obligations due to unforeseen occurrences. Thus, this word is considered

to be negative in financial contexts. If Natural Language Processing applications such as Sentiment Analysis ignore the domain related information, a substantial drop in the performance can occur. Every Natural Language Processing system is domain-specific and requires to account for domain information. The easiest solution to achieve this is to use training data in the domain of interest. However, it entails a large number of human efforts to create labeled training data, which makes the task practically infeasible. To address this issue, previous work has focused on *domain adaptation* – the technique to adapt a model trained on the general domain to a different new domain. In this thesis, we review existing domain adaptation methods, develop a technique for adapting sentiment dictionaries, adapt the models in time considering temporal dynamics in data, and compare and evaluate different approaches for adapting the models in a low-data scenario, which is a typical situation in business and industry.

1.2 PROBLEM STATEMENT

Due to the importance of domain adaptation in Natural Language Processing, this task has become a popular and promising area in computational linguistics. Different algorithms have been developed to investigate this problem. This thesis aims at connecting and systematizing these algorithms discussing their advantages and limitations. Thus, we review a wide range of domain adaptation approaches including the recent advances in this field and the current research.

To understand domain adaptation, we conduct several experiments from different perspectives. First, we look at the domain adaptation in finance. Central interest in finance is identifying the economic drivers of financial outcome variables. An example of such a financial variable is the stock return. The textual information about companies, stock exchanges and the market has a direct impact on the future stock returns. So, the observation of this information is very important for individual investors and large trading institutes. To evaluate such information automatically, financial research uses text analysis based on sentiment dictionaries. The problem with this approach is that the financial domain has a specific vocabulary. A word categorization scheme of sentiment dictionaries derived for the general domain might not translate effectively into a finance domain with its dialect. For instance, words like “cost”, “tax” and “liability” identified as negative by the general dictionary are typically not considered negative in financial contexts. Thus, we need the adaptation of sentiment dictionaries from generic lexicons. In our work, we develop such a method that automatically adapts the general dictionaries for the financial domain.

Second, we perform the adaptation of the models in time. The meaning of the word changes over time and this might significantly

impact the prediction. Especially the language of social media is changing very rapidly – new words appear in the vocabulary, new named entities gain sudden importance and new trends are rapidly emerging. That is why it is important to account for the temporal dynamics in the data, i.e., to adapt the models to new coming data. Social media forums provide a great source of such time-series data. In our work, we use Reddit[1] data for financial forecasting under the consideration of language change over time. In particular, we predict movie sales performing temporal transfer-learning, where the models are trained incrementally by the continuous update.

Third, we evaluate diverse domain adaptation methods in low-data settings, i.e., when there is a lack of training data in the specific domain. This is a typical situation in real-world scenarios where the training data is too restricted and specialized, e.g., in medicine or law. The questions we want to answer are: (1) which domain adaptation approach is more appropriate for such low-data scenarios? (2) what are the conditions under which one approach substantially outperforms the other? To answer these questions, we apply different domain adaptation methods on two tasks, Sentiment Analysis and Part-of-Speech Tagging (POS), in three domains, i.e., Twitter, Movie Reviews and Clinical Domain.

1.3 CONTRIBUTIONS

In our thesis, we perform domain adaptation in the field of finance. A common task in financial research is to evaluate the "tone" of a text, based on the sentiment dictionaries. To consider the specific vocabulary of the financial domain, the adaptation of these sentiment dictionaries is required – the manual construction of a new specific lexicon or automatic adaptation of sentiment dictionaries from generic lexicons. Which approach should be used? We provide evidence that automatic adaptation is better than manual adaptation. In particular, we demonstrate that the automatically domain-adapted dictionary is a more effective predictor of financial outcomes than the manually domain-adapted dictionary. Furthermore, we obtain an insight into the semantics of our dictionaries in our quantitative and qualitative analysis. We find that annotation should be performed based on the word's contexts in the target domain. Otherwise, it can be incorrect due to the expert's a priori belief about a word's meaning.

The growing volume of social media discussions provides a great source of data for natural language based financial forecasting. This data is time series, i.e., data gathered sequentially in time. In our work, we use this data – the discussions from Reddit forums. To our best knowledge, this is the first work that uses Reddit data for financial forecasting considering, besides, temporal dynamics of the

1 https://www.reddit.com/

data. We develop a model that applies transfer learning in a time-dependent manner and apply it to the movie sales prediction task. The algorithm uses a parameter prior based on previous models for estimation at the current time-steps. In a set of experiments, we show that this method allows to successfully transfer knowledge from prior time steps enabling continuous learning without forgetting. We make the following contributions. (i) We demonstrate that our method improves performance when compared to the method that only uses recent data for training. This illustrates the success and importance of parameter transfer from previous models. (ii) We show the benefit of temporal transfer learning over the autoregressive models confirming in this way the value of textual information for financial forecasting. (iii) We achieve better results when compared to the "naive" model that uses data from all time steps for training at once. This verifies the assumption that temporal transfer learning allows us to capture temporal trends in the data by focusing on those features that are relevant in a particular time step, i.e., we obtain more robust models preventing overfitting. Furthermore, unlike the "naive" method, our approach enables us to update already existing models, without the necessity to train a model again from scratch. Therefore, our model can be eventually deployed in a live system for real-time forecasting.

In addition to the domain adaptation problem, the availability of large amounts of high-quality training data is an issue. The performance of models for core Natural Language Processing problems primarily depends on these two factors. That is why it is important to have an appropriate method to perform domain adaptation in low-data settings. In our work we compare and evaluate such methods and study the conditions under which one approach can be better than the other. The experimental results demonstrate that it is possible to successfully perform domain adaptation in low-data settings, e.g., by using knowledge transfer from contextualized deep models. The choice of the contextualized model depends on the task itself – text classification tasks benefit from a model that is pre-trained on a large amount of unlabeled data; for sequence labeling tasks it is sufficient to apply smaller models (e.g., BERT base). These findings demonstrate the value of recently proposed state-of-the-art methods and the importance of selecting an appropriate model for different problems.

1.4 THESIS OUTLINE

This thesis is organized as follows: Chapter 2 describes diverse approaches for domain adaptation, compares them, and discusses their advantages and limitations. Domain adaptation approaches are categorized into three different settings: Section 2.1 summarizes supervised approaches, Section 2.2 discusses semi-supervised approaches,

and Section 2.3 introduces unsupervised approaches. After that, Chapter 3 describes the domain adaptation of sentiment dictionaries for the prediction of two financial outcomes, excess return and volatility. In Chapter 4, we perform financial forecasting in time under the consideration of language change over time. Chapter 5 compares and evaluates domain adaptation methods in low-resource settings. Finally, the Chapter 6 concludes this thesis.

2

BACKGROUND

This chapter reviews and interprets the strategies of domain adaptation describing a large number of approaches and related works. This may help have a better understanding of the current research status and ideas.

2.1 SUPERVISED DOMAIN ADAPTATION

Supervised domain adaptation assumes that there is an access to a large, annotated corpus of data from both a source domain and a target domain. The approach can be divided into augmentation-based and instance-based supervised domain adaptation. Augmentation-based methods transform an original feature space in a new space so that it can be predictive for both domains, and instance-based methods weight individual observations during training based on their importance to the target domain.

AUGMENTATION-BASED METHODS Daumé III (2009) proposes a very simple but effective approach which they call "frustratingly easy" domain adaptation. This method is implemented as a preprocessing step for any supervised learning algorithm. Given a large set of training data in the source domain, and labeled instances in the target domain, the authors augment each feature x_i by forming a source-specific, target-specific and domain invariant version of this feature. Each feature x_i from the source domain is duplicated to three versions: $< x_i, x_i, 0 >$, where the first one refers to a source-specific version, the second one means domain invariant version, and the last one is a target-specific version. In this example, the last element is 0 meaning that x_i is from the source domain. Similarly, each feature x_i from the target domain is formed as $< 0, x_i, x_i >$. After this preprocessing step, the augmented features are fed into a classifier.

The weights of the features demonstrate the intuition behind this method. The algorithm assigns high weight to the domain invariant version, if the feature performs similarly in both domains, whereas it assigns high weight to the source-specific version if this feature is only important for this domain. So, the algorithm can capture the fact that the word "the" is commonly used as determiner across the domains. In contrast, the feature "monitor as a noun" will have a large weight in the general (source) domain, while the feature "monitor as a verb" will have a large weight in the computer (target) domain. This

approach is incredibly simple but demonstrates very good results for many real-world tasks.

INSTANCE-BASED METHODS The problem of instance-based domain adaptation is also known in the literature under the term domain shift: *prior* and *covariate* shift. In this context, different domains correspond to different probability distributions $p(x, y)$ over the same feature-label space pair $X \times Y$, where X is a feature space, and Y is a label space. The target domain is denoted by $p_T(x, y)$ and the source domain is denoted by $p_S(x, y)$. Prior shift refers to a label shift, assuming the conditional distributions remain equal, $p_S(x|y) = p_T(x|y)$ but $p_S(y) \neq p_T(y)$ (Moreno-Torres et al., 2012). In contrast, covariate shift assumes that posteriors remain equal in both domains, $p_S(y|x) = p_T(y|x)$ but $p_S(x) \neq p_T(x)$(Kouw and Loog, 2019) Specifically, the prior shift is a change in the label distribution, and covariate shift appears due to differences in vocabulary and writing style.

In general, instance-based methods minimize the target risk through data from the source domain. The source distribution is related to the target risk as follows (Kouw and Loog, 2019):

$$R_T(h) = \sum_{y \in Y} \int_X l(h(x), y) p_T(x, y) dx \tag{1}$$

$$= \sum_{y \in Y} \int_X l(h(x), y) \frac{p_T(x, y)}{p_S(x, y)} p_S(x, y) dx \tag{2}$$

A loss function l compares a classifier prediction with the true label $l : \mathbb{R} \times Y \longrightarrow \mathbb{R}$, and a function $R_T(h)$ is the expected loss of a target classifier h with respect to a distribution $R(h) = E[l(h(x), y)]$.

Since the posteriors remain equivalent in covariate shift, the joint distributions can be decomposed and the terms can be cancelled (Kouw and Loog, 2019):

$$R_T(h) = \sum_{y \in Y} \int_X l(h(x), y) \frac{\cancel{p_T(y|x)} p_T(x)}{\cancel{p_S(y|x)} p_S(x)} p_S(x, y) dx \tag{3}$$

The ratio of the data marginal distributions, $p_T(x)/p_S(x)$, refers to the importance weighting. A large weight indicates that the sample has high probability under the target distribution. Conversely, the sample has low probability under the source domain. Thus, the loss can increase for certain samples, while decreasing for others. This *instance-based* method was proposed in (Jiang and Zhai, 2007) to handle the covariate shift. The authors demonstrated that incorporating information from the target domain through instance weighting is effective for many NLP tasks.

Since the conditional distributions remain equivalent in prior shift, they can also be cancelled similar to the previous approach (Kouw and Loog, 2019):

$$R_T(h) = \sum_{y \in Y} \int_X l(h(x), y) \frac{\cancel{p_T(x|y)} p_T(y)}{\cancel{p_S(x|y)} p_S(y)} p_S(x, y) dx \tag{4}$$

where $p_T(y)/p_S(y)$ are the class weights, correcting the change in class priors between the domains. This weighting strategy is related to the class imbalance (Jacobusse and Veenman, 2016). Similar effect can also be achieved by under- or over-sampling data points from the class (Chawla et al., 2002).

Instance weighting methods were also studied for Statistical Machine Translation (SMT). For example in (Axelrod, He, and Gao, 2011), the authors extract sentences from a large general domain parallel corpus that are most relevant to the target domain. They choose source domain samples based on the perplexity scores on the language model trained on the target domain data. This method is also known in the literature as *data selection* method. It was also applied in (Tsvetkov et al., 2016) to create improved task-specific word representations. The authors frame this method as a *curriculum learning* – learning the ordering of the training instances, in which the model reads the corpus. They optimize curriculum for training word representations and later use them as the input features in NLP tasks showing the improved performance. The idea behind this method is that some tasks like Part-of-Speech tagging (POS) prefer vectors trained on curricula that promote well-formed sentences. Conversely, Named Entity Recognition (NER) task prefers vectors trained on corpora that begin with named entities (Tsvetkov et al., 2016). So, learning the curriculum helps to improve the performance on downstream tasks over random or natural corpus orders. Inspired by this work, Ruder and Plank (2017) learn data selection measures for transfer learning. They evaluate these measures on domain adaptation task showing the improved performance on several datasets.

2.2 SEMI-SUPERVISED DOMAIN ADAPTATION

Semi-supervised domain adaptation is a problem setting where the labeled data may be too few to build a good classifier. In other words, one has access only to a small amount of labeled data in the target domain. To solve this problem, several semi-supervised approaches have been proposed which make use of different data resources – a large amount of labeled data from a source domain, and a large amount of unlabeled data from both source and target domains.

PRIOR-BASED METHODS Prior-based methods generally assume that there is a small amount of labeled data in the target domain and

a large amount of labeled source data. They perform adaptation during parameter estimation by placing priors over the parameters, i.e., they place a prior distribution over the parameters θ of the model $p(y|x;\theta)$. The idea of this method is that the prior encodes some prior knowledge before the model estimation, and for domain adaptation, the labeled source domain can act as this prior knowledge (Kouw and Loog, 2019). For many learning algorithms such as maximum entropy, SVMs or naive Bayes, the prior can be implemented as a regularizer for the model. Simply put, these algorithms contain a regularization term on the weights θ of the form $\lambda\|w\|_2^2$. In the prior-based methods, this regularization term is replaced by $\lambda\|w - w^s\|_2^2$, where w^s are the parameters learned from the source model and λ specifies regularization strength (larger values means stronger regularization). This allows the model trained on the target data to have weights that are similar to the weights from the source model (Daumé III, 2009).

The first prior-based model was introduced by Chelba and Acero (2006). They use the source model as a prior on the weights for a second one, trained on the target data with maximum entropy classifier. Daumé III and Marcu (2006) perform several experiments demonstrating the benefit of this approach over the baselines for a range of NLP tasks. Finkel and Manning (2009) apply this method for domain adaptation in multi-task setting, so that the performance can be improved across all domains and not a single target domain. They call the model *hierarchical Bayesian domain adaptation* because it makes use of a hierarchical Bayesian prior. In a standard classifier, there will be a parameter (weight) for each feature, and usually, there is a zero-mean Gaussian prior over the parameter values so that they don't get too large. This can be viewed as a Bayesian prior or as weight regularization. The learning process consists of optimizing the log conditional likelihood of the data with respect to the parameters, $L_{orig}(D;\theta)$, where θ are the parameters or feature weights, and D is the data. The model of Finkel and Manning (2009) slightly modifies this function as follows: (i) the model has separate parameters for each feature in each domain, (ii) the model also has a top-level parameter for each feature. For each domain, the Gaussian prior over the parameter values is centered around these top-level parameters (not around zero), and a zero-mean Gaussian prior is placed over the top-level parameters. Thus, each feature weight θ_i is replicated once for each domain, as well as for a top-level set of parameters. The final loss function looks therefore as follows:

$$L_{hier}(D;\theta) = \sum_d (L_{orig}(D_d;\theta_d) - \sum_i \frac{(\theta_{d,i} - \theta_{*,i})^2}{2\sigma_d^2}) - \sum_i \frac{(\theta_{*,i})^2}{2\sigma_*^2} \tag{5}$$

The parameters for domain d are θ_d with individual components $\theta_{d,i}$, the top-level parameters are θ_*, and all parameters collectively are θ.

σ_d^2 and σ_*^2 are variances on the priors over the parameters for all the domains, as well as the top-level parameters.

This framework allows information to be shared between domains and at the same time to override the information from other domains when there is sufficient evidence. The model is essentially equivalent to the domain adaptation approach of (Daumé III, 2009). However, it outperforms the previous approach since the representation of the model conceptually separates some of the parameters which are not separated in (Daumé III, 2009).

2.2.1 *Embedding-based Methods*

Domain generic word embeddings such as Glove (Pennington, Socher, and Manning, 2014) and word2vec (Mikolov et al., 2013b) have demonstrated remarkable success for transferring the prior knowledge to down-stream tasks. These embeddings are usually trained on a corpus such as Wikipedia or Common Crawl. The knowledge transfer is accomplished when they are used as features for supervised learning problems. Since these methods benefit from large amounts of unlabeled data, they can be seen as semi-supervised domain adaptation methods.

To adapt a classifier for a target domain, word embeddings are often computed from scratch on the unlabeled data from the target domain to capture domain-specific semantics and then used as input features to perform classification. For example, *domain specific word embeddings* like Glove Twitter embeddings[1] and biomedical embeddings BioWord2vec (Yijia et al., 2019) have been computed for solving NLP tasks in Twitter and medical domains.

To obtain domain-specific word embeddings of good quality, the large amount of unsupervised data from the target domain should be available (embeddings learned on small data sets are of low quality). However, this is not always the case, especially in industrial and business real-world scenarios. For example, such data as customer support tickets reporting issues, product reviews, reviews of restaurants and movies, discussions by special interest groups and political surveys are usually of a small size (Sarma, 2018). To solve this issue, several approaches have been proposed to adapt existing general-purpose word embeddings to obtain *domain adapted word embeddings*. As usual, they can be used as input features to perform classification.

Sarma (2018) introduces a method for obtaining domain adapted word embeddings that use small-sized supervised data from target domain and the knowledge from generic word embeddings. Domain adapted word embeddings are constructed as follows:

1 https://nlp.stanford.edu/projects/glove/

1. Domain generic embeddings are obtained from algorithms such as Glove or word2vec.

2. Domain-specific embeddings are obtained by applying algorithms such as Latent Semantic Analysis (LSA) on the supervised data from the target domain.

3. Domain generic and domain-specific embeddings are combined via a linear Canonical Correlation Analysis (CCA) (Hotelling, 1936) or a nonlinear kernel CCA (KCCA) (Hardoon, Szedmák, and Shawe-Taylor, 2004). They are projected along with the directions of maximum correlation, and a new domain adapted embedding is obtained by averaging the projected domain generic embeddings and domain-specific embeddings.

Results from the evaluation on sentiment classification tasks demonstrate the efficiency of domain adapted embeddings over domain generic and domain-specific embeddings when used as input features to standard (e.g., bag-of-words) or state-of-the-art sentence encoding algorithms (e.g., InferSent (Conneau et al., 2017)).

A similar approach was proposed in (Bojanowski et al., 2019), i.e., the approach of adapting word vector-based models to new textual data. In contrast to previous work on adaptation to a new domain, they adapt the models in time. The authors claim that the language distribution can drastically change over time and that is why general-purpose pre-trained models require adaptation to fit the distribution of the task at hand. To confirm this fact, they perform several experiments to adapt the models trained on a large corpus to a novel small corpus. The authors formulate this problem as *monolingual word vector alignment* problem. First, they train a model on a small corpus S_1 and obtain the model Y. Then, they find a linear mapping Q that aligns Y and the model X trained on a large corpus S_0. Given the mapped vectors XQ and Y, the final word vectors Z are simply an average

$$z_i = \begin{cases} Q^T x_i & \text{if } i \in V_1 \setminus V_0 \\ \frac{1}{2}(Q^T x_i + y_i) & \text{if } i \in V_0 \cap V_1 \\ y_i & \text{if } i \in V_1 \setminus V_0 \end{cases}$$

where V_0 is the lexicon found in S_0 and V_1 is the lexicon found in S_1.

Using this technique, Bojanowski et al. (2019) adapt word vector models and text classifiers to new data. They show that this approach yields good performance in all setups and outperforms a baseline consisting of fine-tuning the model on new data.

Other approaches suggest to learn *cross-domain word embeddings* using unlabeled data from different domains (Bollegala, Maehara, and Kawarabayashi, 2015; Yang, Lu, and Zheng, 2017). These methods are reregularization based approaches. For example, Bollegala, Maehara,

and Kawarabayashi (2015) relate the source and target word representations via a pivot-regularizer – they consider frequent words in the source domain and the target domain as the "pivots", and then try to use them to predict the surrounding "non-pivot" words, ensuring the pivots to have the same embedding across two domains.

The model of Yang, Lu, and Zheng (2017) also uses a regularization method to perform domain adaptation. The authors implement a regularized skip-gram model, where the source domain information is selectively incorporated for learning the target domain word embeddings. They, first, learn an embedding w_s for each word w from the source domain D_s. Next they learn the target domain embeddings as follows:

$$L'_{D_t} = L_{D_t} + \sum_{w \in D_t \cap D_s} \alpha_w \cdot \| w_t - w_s \|^2 \tag{6}$$

where D_t refers to the target domain, L_{D_t} is the objective of the skip-gram model (Mikolov et al., 2013b), w_t is the representation for w from target domain, and α_w measures the amount of transfer between the two domains. Thus, the learning is accomplished by augmenting the skip-gram objective with this simple regularization term.

Despite the effectiveness of cross-domain word embeddings on various down-stream NLP tasks, this approach appears to be only competitive when a large amount of unlabeled data is available (Yang, Lu, and Zheng, 2017).

2.2.2 *Contextualized Embedding-based Methods*

Recently, contextualized embedding methods have been proposed that go beyond transferring word embeddings. These methods first *pre-train* neural networks on large unlabeled text corpora, and then *fine-tune* the models on downstream tasks. Language modeling has been shown to be an ideal pre-training task for a range of challenging language understanding problems: it captures many language relevant aspects, such as long-term dependencies or hierarchical relations (Howard and Ruder, 2018). Therefore, it helps to learn complex characteristics of a word and take into consideration the context – the vector representation of a word changes with respect to the context in which it appears. These components turned out to be crucial for improving the state of the art across a range of challenging NLP problems, thereby replacing the classic word embeddings.

To adapt the language model for a given task, it has to be trained on the downstream tasks by simply fine-tuning all pre-trained parameters. Since such approach requires minimal task-specific parameters, it can be easily applied for almost any NLP task and domain, ranging from question-answering to sentiment analysis. Thus, this method can also be seen as domain adaptation, i.e., adaptation of pre-trained language model to a target task from the target domain.

LSTM-BASED One of the latest contextualized embedding approaches is an unsupervised language model fine-tuning method (ULMFiT) (Howard and Ruder, 2018). It utilizes the LSTM language model (Hochreiter and Schmidhuber, 1997) for the training process. It consists of general-domain language model pre-training, target task language model fine-tuning and target task classifier fine-tuning. The method is universal in the sense that it works across tasks, it uses a single architecture and training process, it requires no custom feature engineering or preprocessing, and it does not require additional in-domain documents or labels (Howard and Ruder, 2018). Another advantage of ULMFiT is that it utilizes the novel fine-tuning techniques to retain previous knowledge and avoid catastrophic forgetting, for example, gradual unfreezing, discriminative fine-tuning to allow to tune each layer with different learning rates. Howard and Ruder (2018) claim that this technique can prevent overfitting even with only 100 labeled examples and achieve state-of-the-art results also on small datasets. As a result, they outperform the state-of-the-art on different text classification tasks including sentiment analysis, question classification and topic classification.

Another successful LSTM-based contextualized model was introduced in (Peters et al., 2018). The authors call it ELMo (Embeddings from Language Models) representations. Here, word vectors are learned functions of the internal layers of a deep bidirectional language model. Specifically, the model learns a linear combination of the vectors stacked above each input word for each end task. So, first, the bidirectional language model is computed to record all of the layer representations for each word. Next, a linear combination of these representations is calculated. Then, the pre-trained representations are used in task-specific architectures as additional features. This method can be considered as a feature-based approach since all pre-trained parameters used for a downstream task are not fine-tuned, i.e., remain “frozen". Furthermore, to achieve good performance, the model requires the use of task-specific architectures. In terms of domain adaptation, the authors suggest performing fine-tuning of the bidirectional language model on domain-specific data. They claim that it further increases the performance of a downstream task and therefore can be used for domain adaptation.

While contextualized word embeddings are very advantageous for a wide range of NLP tasks, *contextualized string embeddings* (Akbik, Blythe, and Vollgraf, 2018) outperform them on sequence labeling tasks such as Part-of-Speech Tagging (POS) or Named Entity Recognition (NER). Along with the ability to capture word meaning in context, they model words and context as sequences of characters. This allows to better handle rare and misspelt words and model subword structures such as prefixes and endings. Akbik, Blythe, and Vollgraf (2018) set up the sequence labeling architecture as follows: each sen-

tence is passed as a sequence of characters to a bidirectional character-level neural language model, then contextual string embedding is constructed from the internal character states and passed to the BiLSTM-CRF (Huang, Xu, and Yu, 2015) sequence tagging module to address a downstream NLP task.

TRANSFORMER-BASED Later contextualized embedding models utilize Transformer (Vaswani et al., 2017) based language model for pre-training. This model architecture eschews recurrence and instead relies on an attention mechanism to draw global dependencies between input and output. As a result, the methods that use this language model achieve a new state of the art outperforming previously introduced approaches. Among these methods are OpenAI GPT (Radford and Sutskever, 2018), GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2019), XLNet (Yang et al., 2019), BERT (Devlin et al., 2019) and its optimized variants RoBERTa (Liu et al., 2019), and DistillBERT (Sanh et al., 2019). The main difference between all these models is the use of different unsupervised pre-training objective.

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) is one of the latest Transformer based models. While previous models like ELMo (Peters et al., 2018) and GPT (Radford and Sutskever, 2018) use unidirectional language modeling to learn language representations, this method is designed to pre-train bidirectional representations. This allows to model the context in both directions which is crucial for many NLP tasks ranging from sentence-level and question answering tasks to token-level tasks. To train a deep bidirectional representation, the authors use the so-called "masked language model" (MLM) pre-training objective: they mask 15% of the input tokens at random and then predict those masked tokens. The final hidden vectors corresponding to the mask tokens are fed into an output softmax over the vocabulary, as in standard language modeling.

In addition to the masked language model, Devlin et al. (2019) also use a "next sentence prediction" task that jointly pre-trains text-pair representations. This is important for the downstream tasks such as Question Answering (QA) and Natural Language Inference (NLI) which are based on understanding the relationship between two sentences. Language modeling alone can not directly capture this information. To achieve this, they use pairs of sentences with 50% of pairs with actual next sentences (labeled as IsNext), and other 50% of pairs with a random next sentence (labeled as NotNext). Thus, the use of two unsupervised tasks for pre-training allows BERT to handle a variety of down-stream tasks.

The input embeddings for BERT are the sum of different embeddings: the token WordPiece embeddings (Wu et al., 2016) with a 30,000 token vocabulary, the position embeddings, and the segmenta-

tion embeddings for sentence pair tasks indicating to which sentence a certain token belongs. The advantage of using wordpieces is the need for special treatment of unknown words. Furthermore, it gives a good balance between the flexibility of single characters and the efficiency of full words for decoding (Wu et al., 2016). Position embeddings are necessary to identify the relative positions of each token in the sentence. Finally, segmentation embeddings help to distinguish between two sentences for sentence pair tasks.

For fine-tuning, the BERT model is initialized with the pre-trained parameters, and all of the parameters are adapted using labeled data from the downstream tasks. At the output, the token representations are fed into an output layer For token level tasks, the token representations are fed into an output layer, and for classification, the [CLS] (a special symbol in front of every input example) representation is fed into an output layer.

Due to these novel techniques introduced in (Devlin et al., 2019), this model has achieved state-of-the-art performance in many NLP tasks ranging from sequence classification and sequence-pair classification to question answering.

Later, several models have been introduced that try to eliminate certain issues of the BERT model. For example, Yang et al. (2019) claim that the [MASK] token used for pre-training is absent from real data at the fine-tuning time, resulting in a pre-train-fine-tune discrepancy. Besides, since the predicted tokens are masked in the input, BERT is not able to model the joint probability using the product rule. So, Yang et al. (2019) propose XLNet, a generalized autoregressive method that avoids these limitations. It uses the permutation operation so that all tokens can be predicted but in random order (in BERT's model only the masked (15%) tokens are predicted). This allows to not rely on data corruption avoiding pre-train-fine-tune discrepancy. Moreover, XLNet uses the product rule for modeling the joint probability of the predicted tokens eliminating the independence assumption made in BERT (Yang et al., 2019). In the pre-training framework, it furthermore uses the state-of-the-art language model, Transformer-XL (Dai et al., 2019) which enables learning dependency beyond a fixed length. All these components improve the performance especially for tasks involving a longer text sequence. Hence, the authors demonstrate consistent improvement over BERT on a wide spectrum of problems including language understanding, reading comprehension text classification, and document ranking tasks.

Another optimized BERT model is RoBERTa (robustly optimized BERT) (Liu et al., 2019). The authors improve the BERT model by simple but effective modifications: they train the model longer, on longer sequences, with bigger batches and over more data. Moreover, they remove the next sentence prediction objective and introduce dy-

namic masking so that the masked token changes during the training epochs. As a result, RoBERTa outperforms both BERT and XLNet on several language understanding tasks.

While the above models lead to significant improvement, they often have several hundred million parameters which makes it difficult to run on the edge, e.g. on mobile devices. For this reason, Sanh et al. (2019) introduced a smaller, faster and lighter version of BERT – DistillBERT (a distilled BERT). The authors claim that this model still retains 97% of BERT language understanding capabilities, being 60% faster and reducing the size of a model by 40%. DistillBERT eliminates the computational and memory issues by leveraging a compression technique, knowledge distillation, in which a compact model is trained to reproduce the behavior of a larger model. DistillBERT has essentially the same general architecture as BERT while removing token-type embeddings, the pooler and retaining only one half of layers. It furthermore applies best practices of RoBERTa, i.e., dynamic masking, removing next sentence prediction and using very large batches.

BERT-BASED DOMAIN ADAPTATION While contextualized word embeddings and their further fine-tuning became a widely used method to transfer knowledge from the general domain to a new domain, Han and Eisenstein (2019) claim that the applicability of this approach is still unknown when the target domain varies substantially from the pre-training corpus. That is why they distinguish between *task specific fine-tuning* and *domain-adaptive fine-tuning*. The authors claim that task-specific fine-tuning method may help to adapt the contextualized embeddings to a new labeling task, but not to the domain. So, they propose domain-adaptive fine-tuning model *AdaptaBERT*, which adds masked language modeling objective over unlabeled text in the target domain. Specifically, before applying task-specific fine-tuning, they use BERT training procedure to a dataset that includes all available target domain data. In addition to this data, they also use an equal amount of unlabeled data from the source domain. This results in significant improvements on sequence labeling tasks from a specific domain like Twitter or Early Modern English.

Similar approaches were proposed in (Xu et al., 2019) and (Rietzler et al., 2020), where the language model was first fine-tuned on domain-specific corpora. Both systems evaluate the method on aspect-based Sentiment Analysis showing substantial improvement over vanilla BERT-base and XLNet-base models. Rietzler et al. (2020) furthermore builds cross-domain adapted BERT language model which performs even better than a BERT-base model that is trained in-domain.

Another BERT-based domain adaptation framework was proposed in (Ma et al., 2019). They employ the idea of domain-adversarial training which is executed in two separate steps: (1) a BERT-based domain

classifier is trained on data from different domains with domain labels. Here, the probability scores from the domain classifier quantify the domain similarity. (2) the domain probabilities are then used for curriculum training whose idea is to learn from easy samples first, i.e., from samples similar to the target domain data.

The main characteristic of all proposed BERT-based domain adaptation methods is the availability of a large amount of unsupervised target data. This might be an issue in real-world business and industrial scenarios where there is not enough target data, either supervised or unsupervised.

2.3 UNSUPERVISED DOMAIN ADAPTATION

The major idea in unsupervised domain adaptation is to learn a domain invariant representation which can be used to train the classifier on the labeled data from the source domain and apply it on the target domain. Domain invariant representations can be learned to leverage both labeled data from the source domain and unlabeled data from the source and target domains. This problem setting has also been referred in the literature as "domain adaptation without target labels" (Kouw and Loog, 2019) and "transductive transfer learning" (Pan et al., 2010a).

AUGMENTATION-BASED METHODS Blitzer, McDonald, and Pereira (2006) introduce *structural correspondence learning (SCL)* domain adaptation method that exploits unlabeled data from both source and target domains. This method is based on the idea of finding a common feature representation that is meaningful across domains using pivot features. The authors define these features on the unlabeled data considering the following conditions: pivot features should occur frequently in both domains, and they must be diverse enough to adequately characterize the nuances of the supervised task. For example, determiners are good pivot features for Part-of-Speech tagging, since they occur frequently in any domain, but choosing only determiners will not help to discriminate between nouns and adjectives. In Figure 1, the words in italics are pivot features because they have the same part-of-speech tags in both domains, and are also indicative for the POS tags of the non-pivot features, i.e, the words in bold. For example, if "required" appears to the right of a word, then that word is likely to be a noun. This example captures the intuition behind the SCL method. Pivot features are used to put domain-specific words in correspondence. Simply put, the pivot features model the fact that in the biomedical domain, the word "signal" behaves similarly to the words "investments", "buyouts" and "jail" in the financial news domain.

(b) MEDLINE occurrences of signal, together with pivot features	**(c)** Corresponding WSJ words, together with pivot features
the **signal** *required* to stimulatory **signal** *from* essential **signal** *for*	of **investment** *required* of **buyouts** *from* buyers to **jail** *for* violating

Figure 1: Corresponding words are in bold, and pivot features are italicized (Blitzer, McDonald, and Pereira, 2006).

After defining m pivot features, Blitzer, McDonald, and Pereira (2006) build m binary classifiers for each of the pivot features to model the correspondence between them and the non-pivot features. An example of such binary classification problem is of the form "Is <the token on the right> required?". Next, they perform Singular Value Decomposition (SVD) on the joint weight matrix W of these classifiers and obtain a projection matrix θ. Then, they project the original feature space X into a new space, θX. Finally, they augment the original features with the transformed features and train a classifier on the source domain using both the original and transformed feature versions.

If non-pivot features from two domains are highly correlated with the same pivot features, then they will be projected to the same space in the latent space. So, if the learned mapping θ is of good quality, then the classifier learned on the source domain can be effective on the target domain. Hence, this allows performing SCL-based domain adaptation in an unsupervised fashion.

DOMAIN-INVARIANT FEATURE-BASED METHODS

Autoencoder-based Methods Autoencoder-based methods showed promising results for obtaining robust representations. *Autoencoder (AE)* is an artificial neural network for transforming the original feature space, where the outputs are set to x, the inputs (Le, Patterson, and White, 2018). Autoencoders are designed to be unable to learn to reconstruct the input perfectly – they are restricted in ways that allow them to copy only approximately (Goodfellow, Bengio, and Courville, 2016). Thus, the model is forced to prioritize which aspects of the input should be copied. That is why the autoencoder can learn the underlying and useful properties of the data.

The autoencoder architecture consists of two parts: an encoder $h = f(x)$ and a decoder that produces a reconstruction $r = g(h)$. This architecture is presented in Figure 2. Autoencoders may be viewed as a special case of feedforward networks trained with all the same techniques, e.g., minibatch gradient descent (Goodfellow, Bengio, and Courville, 2016).

One way to obtain useful representations from the autoencoder is to constrain h to have a smaller dimension than x. Such an autoen-

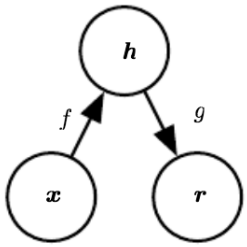

Figure 2: The general structure of an autoencoder, mapping an input x to an output r through an internal representation h. The autoencoder has two components: the encoder f (mapping x to h) and the decoder g (mapping h to r) (Goodfellow, Bengio, and Courville, 2016).

coder is called *undercomplete*. The learning process is simply minimizing a loss function (Goodfellow, Bengio, and Courville, 2016):

$$L(x, g(f(x))) \tag{7}$$

where L is a loss function penalizing $g(f(x))$ for being dissimilar from x, such as the mean squared error.

Another way to obtain useful features from the autoencoder is to use *denoising autoencoders (DAE)*, a variant of the AE method for reconstructing the input vectors from stochastically corrupted input signals (Vincent et al., 2008). A denoising autoencoder minimizes (Goodfellow, Bengio, and Courville, 2016):

$$L(x, g(f(\grave{x}))) \tag{8}$$

where $\grave{x}$ is a copy of x that has been corrupted by some noise. Denoising autoencoders must, therefore, undo this corruption rather than simply reconstruct the input. Thus, denoising autoencoders can also learn the valuable properties of the data. Denoising autoencoder training procedure is demonstrated in Figure 3. The autoencoder learns a reconstruction distribution $p_{reconst(x|\grave{x})}$ from training pairs $(x, \grave{x})$ as follows:

- A training example x is sampled from the training data.
- A corrupted version $\grave{x}$ is sampled from $C(\grave{x}|x = x)$
- $(x, \grave{x})$ is used as a training example for estimating $p_{reconst}(x|\grave{x}) = p_{decoder}(x|h)$ with h the output of encoder $f(\grave{x})$ and $p_{decoder}$ defined by a decoder $g(h)$.

One can simply perform gradient-based approximate minimization on the negative log-likelihood-log $p_{decoder}(x|h)$. The denoising autoencoder is a feedforward network and may be trained with the same techniques as any other feedforward network. Typical choices of

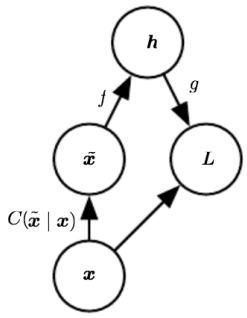

Figure 3: The general structure of denoising autoencoder, which is trained to reconstruct the data point x from its corrupted version $\grave{x}$. This is done by minimizing the loss $\mathrm{L} = -\log \mathrm{p}_{\mathrm{decoder}}(x|\mathrm{h} = \mathrm{f}(\grave{x}))$, where $\grave{x}$ is a corrupted version of the data example x. $\grave{x}$ is obtained through a corruption process $\mathrm{C}(\grave{x}|x)$. (Goodfellow, Bengio, and Courville, 2016).

corruption include typically Gaussian noise or masking noise (Chen et al., 2015).

Once a DAE has been trained, one can stack another DAE on top of it, by training a second one which takes as an input the encoded output of the first one (Glorot, Bordes, and Bengio, 2011). This architecture is known as *Stacked Denoising Autoencoder (SDA)* (Vincent et al., 2008). Once the SDA is trained, their parameters describe multiple levels of representation for x and can be used to initialize a supervised deep neural network (Bengio, 2009). Glorot, Bordes, and Bengio (2011) propose domain adaptation method based on this method. They train SDA on the unlabelled data from all domains to build effective feature representations and then train a linear classifier on the transformed labeled data of the source domain. The corruption process here is a masking noise, i.e. each active input has a probability P to be set to 0. They also add a Gaussian corruption noise, which is added before the activation function of the input layer in order to keep the sparsity of the representation. To evaluate the model, they test the resulting classifier on the target domain. This unsupervised method shows good results for performing domain adaptation for sentiment classification, which also scales well and allows to use it on a dataset of many different domains. The authors make an observation that if multiple domains are available, sharing the unsupervised pre-training of SDA is advantageous compared to pre-training on the source and target only. A good example is the classification of product reviews according to their sentiments. Assume that we have the source data consisting of *book reviews* and the target data consisting of *kitchen appliances*. A classifier trained on the source domain never sees the bigram "energy efficient" during training and thereby assigns zero weight to it. In the SDA representation, this bigram would typically be reconstructed by similar sentiment, such as

"good" or "love". Thus, the classifier trained on the source data can assign weights even to features that never occur in its original representation (Chen et al., 2015).

Chen et al. (2012) significantly speeds up the above approach by proposing the *marginalised SDA (mSDA)* model, which marginalizes noise and thus does not require stochastic gradient descent or other optimization algorithms to learn parameters. Their method achieves performance on domain adaptation comparable to the traditional SDAs while reducing the training time significantly. Later, Yang and Eisenstein (2014) present a new version of mSDA by incorporating alternative noising technique which they call *marginalized structured dropout*. In many NLP settings, there are several feature templates like previous-word, middle-word, next-word, etc. Yang and Eisenstein (2014) exploit this structure by using an alternative dropout scheme: for each token, they choose exactly one feature template to keep and zero out all other features that consider this token. This further increases the training speed over previous work and yields state-of-the-art accuracy when applied to the domain adaptation task of fine-grained part-of-speech tagging.

Domain-adversarial-based Methods Adversarial training is a framework for estimating the models via an adversarial process, in which two models are trained simultaneously: a generative model G, and a discriminative model D that estimates the probability that a sample came from the training data rather than G (Goodfellow et al., 2014). The training procedure for G is to maximize the probability of D making a mistake. In other words, the discriminator gets real and generated examples and tries to distinguish between them, while the generator tries to create examples that are hard to distinguish from real data. This model is known as *Generative Adversarial Network (GAN)*.

The adversarial setting is interesting in the context of regularization because one can reduce the error rate on the test set via adversarial training – training on adversarially perturbed examples from the training set (Goodfellow, Bengio, and Courville, 2016). In the context of domain adaptation, adversarial training is based on the idea of learning domain-invariant representations by encouraging domain confusion through an adversarial objective. The aim of the *Adversarial Domain Adaptation (ADA)* is thereby to learn such representations by reducing the domain discrepancy. Inspired by GAN, Ganin et al. (2016) implement such a non-generative domain-adversarial network for Natural Language Processing. They call this model *Domain-Adversarial Neural Network (DANN)* that uses standard layers and loss functions and can be trained using standard backpropagation based on stochastic gradient descent. The only non-standard component of this architecture is a *gradient reversal layer* that reverses the gradient by multiplying it by a negative scalar during the backpropagation. The

proposed architecture includes: (i) a deep feature extractor, (ii) a deep label predictor, which together with the feature extractor forms a standard feed-forward architecture, (iii) a domain classifier connected to the feature extractor via a gradient reversal layer that multiplies the gradient by a negative constant during the backpropagation. Otherwise, the training standardly minimizes the label prediction loss (for source examples) and the domain classification loss (for all samples). Gradient reversal ensures that the feature distributions over the two domains are as indistinguishable as possible for the domain classifier, thus resulting in the domain-invariant features. Thus, the proposed architecture allows obtaining representations that are discriminative for the learning task and domain-invariant with respect to the shift between domains. This method demonstrated the success for sentiment analysis and image classification outperforming autoencoder-based approaches, e.g., mSDA of (Chen et al., 2012).

Adversarial domain adaptation was also successfully used for cross-language adaptation (Chen et al., 2018). The authors tackle the sentiment classification problem in low-resource languages. They propose an architecture to transfer the knowledge learned from labeled data on a resource-rich source language to low-resource languages where only unlabeled data exists. Their framework has two components: a sentiment classifier and an adversarial language discriminator. Both components take input from a shared feature extractor to learn hidden representations that are simultaneously indicative for the classification task and invariant across languages.

Domain adaptation in adversarial setting was also beneficial for other Natural Language Processing tasks, such as retrieval-based question answering (Yu et al., 2018), question-question similarity (Shah et al., 2018), and representation learning (Shen et al., 2018a).

SUMMARY In this Chapter, we reviewed several domain adaptation methods, which can be classified into three different settings: supervised, semi-supervised, and unsupervised. Most previous works focus on the former two settings due to the lack of training data in specialized domains.

The domain adaptation approaches can also be classified into five contexts based on "how to perform domain adaptation". They include the methods: augmentation-based, instance-based, prior-based, domain-invariant feature-based, and embedding-based. Augmentation-based methods transform an original feature space so that it can be predictive for both domains. Instance-based methods weight individual observations during training based on their importance to the target domain. Prior-based methods perform adaptation during parameter estimation by placing priors over the parameters. Domain-invariant feature-based methods learn domain invariant representations such that a source classifier performs well on the target domain,

and embedding-based methods rely on embeddings, either domain-specific, cross-domain, and domain adapted word embeddings, or contextualized embeddings, which capture syntactic and semantic knowledge about the language that is crucial for domain adaptation.

All in all, our categorization gives a good overview of the existing ideas and current research status on domain adaptation and raises questions that might be important to further research.

3

AUTOMATIC DOMAIN ADAPTATION OUTPERFORMS MANUAL DOMAIN ADAPTATION FOR PREDICTING FINANCIAL OUTCOMES

3.1 INTRODUCTION

A common method employed by finance and accounting researchers is to examine the "tone" and sentiment of various data sources such as corporate 10-K reports, newspaper articles, press releases, and investor message boards (Loughran and Mcdonald, 2011). 10-K reports, also known as 10-K filings, is an interesting resource of textual information. They are the companies' annual disclosures that contain financial statements and information about business strategy, risk factors, and legal issues (Sedinkina, Breitkopf, and Schütze, 2019). Companies are required by law to submit these reports according to the mandate of the U.S. Securities and Exchange Commission (SEC). Such disclosure in forms and documents is necessary to ensure that adequate information is available to investors.

The results of the analysis of these textual sources (Antweiler and Frank, 2004; Tetlock, Saar-Tsechansky, and Macskassy, 2008) indicate that negative word classifications can be effective in measuring tone. This analysis usually relies on the Harvard Psychosociological Dictionary, specifically, on the Harvard-IV-4 TagNeg (H4N) word list[1]. Loughran and Mcdonald (2011) question the use of this lexicon in the financial environment. They argue that word lists developed for psychology and sociology misclassify common words in the financial text. They analyze a large sample of 10-Ks from 1994 to 2008 and find that almost three-fourths of the words identified as negative by the Harvard Dictionary are words typically not considered negative in financial contexts. Many words (e.g., "cost", "liability" and "tax") are in fact neutral or even positive in the financial domain. To overcome this issue, they manually develop an alternative negative word list which reflects better the sentiment in financial text. We will refer to this sentiment dictionary as L&M. To evaluate whether these word lists gauge tone, Loughran and Mcdonald (2011) use them to predict financial variables such as 10-K filing returns and return volatility.

In our work, we also create sentiment dictionaries using 10-Ks, but we adapt them for the finance domain *automatically*. In our experiments, we demonstrate that automatically adapted word lists perform better at predicting financial variables than manually created dictionaries of Loughran and Mcdonald (2011). Our word lists also

1 http://www.wjh.harvard.edu/~inquirer/homecat.htm

outperform the financial dictionary of Theil, Stajner, and Stuckenschmidt (2018) which was created by automatically extending Loughran and Mcdonald (2011) lexicon. To shed light on the superior performance of our sentiment dictionary, we also perform an analysis of the differences between the classifications of L&M and those of our dictionary.

In our experiments, we use sentiment dictionaries and ordinary least squares (OLS), an established method in empirical finance. The choice of this method is motivated by the fact that the classification decision in finance must be interpretable and statistically sound. We can analyze the significance, effect size, and dependence between predictor variables. By looking at the dictionary words occurring in a document, we can furthermore trace the classification decision back to the original data and, e.g., understand the cause of a classification error (Sedinkina, Breitkopf, and Schütze, 2019).

In our study, we learn three main lessons that can also be applied to many other areas. First, interpretable classification decisions are more likely to be trusted than those by black boxes. Second, NLP applications are domain-specific and therefore require domain adaptation to achieve a good performance. Applications based on lexicons require the adaptation of these lexicons as well. Such domain-specific lexicons can be built manually from scratch or automatically adapted from generic lexicons. In our work, we demonstrate that automatic adaptation works better. Third, words often have specific meanings that can be recognized in a context. If only the generic meaning is present to the annotator, the risk of misclassification can be very high. In our analysis, we show that this is the main issue of manual lexicons in our experiments. Thus, the annotation should be performed based on the word's *contexts in the target domain*, not in isolation.

3.2 RELATED WORK

Finance and accounting researchers traditionally used the H4N TagNeg dictionary[2] to examine the tone of a text. This dictionary's sentiment classifications are intended for applications in psychology and sociology. It includes 85,221 words, 81,033 of which are labeled "common". The remaining words are labeled "negative". Loughran and Mcdonald (2011) find that many words from the Harvard list are not negative in a financial context. For example, words such as "tax", "cost", "foreign", "liability", and "depreciation" appear with great frequency in 10-Ks but do not typically have a negative sentiment. Other words like "cancer", "tire", or "capital" rather identify a specific industry segment in 10-Ks than a negative financial event. So, Loughran and Mcdonald (2011) examine all words in 10-Ks that occur in at least 5% of documents and create manually a list of words with negative

2 http://www.wjh.harvard.edu/~inquirer/

sentiment in financial contexts. Furthermore, they expand the word classification categories. In addition to the negative word lists, they consider positive, negative, uncertainty, litigious, constraining, superfluous, interesting, modal, and irregular verb word categories. The constraining class was introduced to measure financial constraints. Therefore, this category has a narrower thematic focus than the other categories. The classes superfluous, interesting, and irregular verb are secondary categories.

The litigious list categorizes words reflecting a propensity for the legal contest. The list includes 731 words such as "claimant", "deposition", "interlocutory", "testimony", and "tort". Uncertainty word list comprises words denoting uncertainty, with emphasis on the general notion of imprecision. The list consists of 285 words, such as "approximate", "contingency", "depend", "fluctuate" and "variability". Negative word list includes 2,337 words like "loss", "termination", "against" and "default". A positive word list was included more in the interest of symmetry than in an expectation of discerning an impact on sentiment identification. This word list has 353 words such as "achieve", "attain", "efficient", "improve" and "profitable". L&M then demonstrate a significant relationship between the occurrence of words from one of their dictionaries in a company's 10-K and financial outcomes like excess return and volatility of the company's stock. For instance, they show that firms using fewer negative and uncertain words receive a more positive reaction from the market after the 10-K filing date. They also conclude that a higher proportion of negative words is linked to larger volatility in the year after the filing. In contrast, L&M found no such consistent relationship for the original H4N dictionary. L&M, therefore, suggest using their dictionaries to gauge sentiment in financial research. Building on L&M, Tsai and Wang (2014), and Theil, Stajner, and Stuckenschmidt (2018) show that the L&M dictionaries can be further improved by adding the most similar words according to the embedding model.

A large part of the work has focused on domain adaptation of dictionaries. We distinguish three main method categories (Sedinkina, Breitkopf, and Schütze, 2019):

Seed-based methods are unsupervised methods that employ seed words at first, and then learn sentiment of words based on different patterns in unlabeled corpora. Some models consider syntactic structure patterns (Hatzivassiloglou and McKeown, 1997; Widdows and Dorow, 2002), or rely on co-occurrence information (Igo and Riloff, 2009; Turney, 2002). Other models use word co-occurrences to construct lexical graphs to perform label propagation over these graphs (Huang, Niu, and Shi, 2014; Velikovich et al., 2010).

Supervised methods use a large training set, not just a few seeds. For instance, Mohammad, Kiritchenko, and Zhu (2013) constructed word lists by calculating PMI between the word and sentiment labels

(emoticons) in tweets. Further methods improved this approach by using distributed word representations (Amir et al., 2015; Rothe, Ebert, and Schütze, 2016; Tang et al., 2014; Vo and Zhang, 2016). For example, Tang et al. (2014) incorporated in word embeddings document-level sentiment supervision. Later, Wang and Xia (2017) integrated the sentiment supervision at both document and word levels. Hamilton et al. (2016) utilized in its turn domain-specific word embeddings and a label propagation approach to conduct the dictionary. They claim that they obtain high-quality word lists, in particular for the finance domain.

Dictionary-based approaches use hand-curated lexical resources – usually WordNet (Fellbaum, 1998) – for dictionary construction (Baccianella, Esuli, and Sebastiani, 2010; Takamura, Inui, and Okumura, 2005; Vicente, Agerri, and Rigau, 2014). Hamilton et al. (2016) argue that dictionary-based approaches generate higher-quality lexicons, due to their use of these clean, hand-curated resources. In our work, we compare two ways of using a hand-crafted resource – an automatically adapted general-domain resource vs. a manually created resource for the specific domain – and show that automatic domain adaptation performs better.

Apart from domain adaptation of dictionaries, other methods of **generic domain adaptation** have been proposed. Most of this work focuses on the supervised domain adaptation: there is a large labeled training set in the source domain and a small amount of labeled target data that is insufficient to achieve good performance (Blitzer, McDonald, and Pereira, 2006; Chelba and Acero, 2006; Chen et al., 2012; Daumé III, 2009; Glorot, Bordes, and Bengio, 2011; Pan et al., 2010b). More recently, domain-adversarial training was successfully applied for domain adaptation (Ganin et al., 2016).

In contrast to this work, we employ a domain adaptation of sentiment dictionaries. We do not transfer any parameters from source to target domain. Instead, we use a general-domain lexicon and adapt it to a specialized domain. First, we create domain-specific embedding vectors, then we train a classifier on source domain labels to obtain a new sentiment dictionary, and finally we build a regression model that is trained using this dictionary. Training target embeddings with source labels gives good results because the divergence between source and target sentiment labels is relatively minor.

Since our work primarily addresses a finance application, we use a model based on sentiment dictionaries. This allows us to provide explanations of the model's decisions and predictions.

3.3 METHODOLOGY

3.3.1 *Empirical finance methodology*

One way to evaluate the quality of sentiment dictionaries is to demonstrate the correlation between financial variables (e.g., excess return, volatility) and the occurrence of negative words in 10-Ks. To achieve this, we use an empirical finance method, Ordinary Least Squares (OLS). This method allows predicting a dependent variable like excess return based on a linear combination of explanatory variables. In our work, we want therefore to examine whether our text-based variables can be helpful for the prediction of financial variables like excess return or volatility.

The methodology is to estimate regression coefficients which represent the explanatory power of text-based variables. Since the variation in the dependent variable might be explained by several regressors, it is necessary to include so-called control variables such as firm size and book-to-market ratio. They allow control for their influence and therefore isolate the *complementary information*. In this way, we can assess the benefit of text-based variables.

We use sentiment dictionaries to create text variables. The value of each text variable is the proportion of tokens in the 10-K that are contained in the dictionary. For example, if the 10-K is 10000 tokens long and 10 of those tokens are contained in the L&M negative dictionary, then the value of the L&M negative text variable for this 10-K is 0.001.

The value of the text variable for a dictionary D is the percentage of tokens from D that occur in a 10-K.

In finance applications, there are two general forms of dependence in the data. Time-series dependence assumes that the residuals of a given firm may be correlated across years for a given firm (*firm effect*). Alternatively, cross-sectional dependence suggests that the residuals of a given year may be correlated across different firms (*time effect*). To address this dependence issue, we model data with both firm and time effects and run an OLS regression with standard errors, which are clustered on two dimensions (Gelbach and Miller, 2009). We refer to this regression as *two-way robust cluster regression* which is as clustering across time within a firm and across firms within a given period. Loughran and Mcdonald (2011) use the method of Fama and MacBeth (1973) instead. This method only considers cross-sectional regression, i.e., yearly estimates of the coefficient are independent of each other. However, this is not true when there is a firm effect in the data.

We apply this regression-based methodology to examine to which extent different financial dictionaries can predict excess return and volatility. This method enables us to compare the explanatory power

of sentiment dictionaries and test the hypothesis that negative words are correlated with lower excess returns and higher volatility.

3.3.2 *Excess Return*

The dependent variable excess return is defined as the firm's buy-and-hold stock return minus the CRSP[3] value-weighted buy-and-hold market index return over the 4-day event window started on the 10-K filing date expressed as a percentage.

In addition to the independent text variables, we include the following financial control variables. (i) Firm size: the log of the number of shares outstanding times the price of the stock. (ii) Alpha of a Fama-French regression (Fama and French, 1993) calculated using days [-252 -6];[4] this represents the "abnormal" return of the asset, i.e., the part of the return that is not explained by the common risk factors such as market and firm size. (iii) Book-to-market ratio: the log of the book value of equity divided by the market value of equity. (iv) Share turnover: the volume of shares traded in days [-252 -6] divided by shares outstanding on the filing date. (v) Earnings surprise, computed by IBES from Thomson Reuters[5]; this variable shows whether the reported financial performance was better or worse than expected.[6]

3.3.2.1 *Volatility*

Return volatility is defined as the root-mean-square error (RMSE) of a Fama-French regression from days [6 252], i.e., 252 days following the filing date, with the first 5 days following the filing date excluded. The RMSE characterizes the stock price variation that cannot be explained by the common risk factors of the Fama-French model. It measures, therefore, the financial uncertainty of the firm. In addition to the independent text variables, we include the following financial control variables. (i) Pre-filing RMSE and (ii) pre-filing alpha of a Fama-French regression calculated from days [-252 -6]; these values capture the return volatility and abnormal return of the firm in the past (see 3.3.2 for alpha and the first sentence of this section for RMSE). (iii) Filing abnormal return; the value of the buy-and-hold return in trading days [0 3] minus the buy-and-hold return of the market index. (iv) Firm size and (v) book-to-market ratio (the same as in 3.3.2). (vi) Calendar year dummies and Fama-French 48-industry dummies to

3 http://www.crsp.com

4 [-252 -6] stays for the 252 days prior to the filing date with the last 5 days prior to the filing date excluded.

5 http://www.thomsonreuters.com

6 Our setup is similar but is not identical to the one used by Loughran and Mcdonald (2011). Our estimates use a sample over a different period (1994-2013) compared to L&M work (1994-2008). Furthermore, not all data used by Loughran and Mcdonald (2011), are publicly available.

dictionary	size
neg_{lm}	2355
unc_{lm}	297
lit_{lm}	903
neg_{ADD}	2340
unc_{ADD}	240
lit_{ADD}	984
neg_{RE}	1205
unc_{RE}	96
lit_{RE}	208
$H4N_{ORG}$	4188
$H4N_{RE}$	338

Table 1: Number of words per dictionary (Sedinkina, Breitkopf, and Schütze, 2019).

allow for time and industry fixed effects.[7] These variables control for unobserved effects. For example, each entity has its industry characteristics that may or may not influence the predictor variables. So, this time-invariant property is explicitly included as a control variable to avoid an omitted variable bias of the estimated coefficients.

3.3.3 *NLP Methodology*

We pose two research questions in our work:

Q1. Which sentiment dictionary is a more effective predictor of financial variables – manually adapted or automatically adapted?

Q2. L&M manually reclassified H4N words to adapt them for the financial domain and showed that this alternative dictionary is more effective than H4N for the prediction of financial outcomes. Can we further improve L&M's sentiment dictionary by automatic domain adaptation?

For domain adaptation, we employ the methodology based on word embeddings – we train word2vec (Mikolov et al., 2013a) using 10-Ks corpus (see 3.4 for details). We consider two adaptations: ADD and RE.

ADD. This method extends the existing L&M dictionary. Each word is represented as its embedding. Training set comprises of L&M words that are labeled +1 if they are marked for the category by L&M and labeled -1 otherwise (where the category is negative, uncertain, or liti-

7 We do not include in the regression a Nasdaq dummy variable showing whether the firm is traded on Nasdaq. Nasdaq mainly lists tech companies, so the Nasdaq effect is already captured by industry dummies.

gious). The test set is the words from the H4N dictionary that are not contained in the L&M dictionary. We also ignore those H4N words that do not have embeddings because their frequency is below the word2vec frequency threshold.

SVM scores are transformed into probabilities via logistic regression. A word that is not in D (e.g., L&M dictionary) is added to D′ if its SVM score is greater than a confidence threshold of θ. This threshold ensures that the words in the adapted dictionary D′ are reliable indicators of the category of interest.

RE. We train SVMs as for ADD, but instead of using a test set, we perform five-fold cross-validation. Again, SVM scores are transformed into probabilities via logistic regression. A word w from D is added to D′ if its converted SVM score of the SVM that was not trained on the fold that contains w is greater than θ.

To answer our first question Q1: "Is automatic adaptation better than manual adaptation?", we apply the adaptation method RE to H4N dictionary and compare the results to the L&M dictionaries.

To answer our second question Q2: "Can L&M dictionary be further improved by automatic adaptation?", we apply adaptation methods RE and ADD to the L&M dictionaries and compare the results for the original L&M dictionaries:

(i) negative (abbreviated as "neg")

(ii) uncertain (abbreviated as "unc")

(iii) litigious (abbreviated as "lit")

Table 1 gives dictionary sizes.

3.4 EXPERIMENTS AND RESULTS

In our work, we used the collection of documents of companies who are required by law to file forms with the U.S. Securities and Exchange Commission (SEC). Since 1934, this commission has requested disclosure in forms and documents to improve how investors find and use information. In 1984, Electronic Data Gathering, Analysis, and Retrieval system (EDGAR)[8] began collecting electronic documents to help investors to get this information. So, the companies submit their filings on EDGAR providing distinct information such as registration statements, insider trading reports, quarterly and annual reports. We downloaded these files from EDGAR, i.e. 206,790 annual SEC 10-K filings for years 1994 to 2013. During the preprocessing procedure, we remove the table of contents, page numbers, links, and numeric tables. Sections that are not useful for textual analysis are deleted as well. Thus, we consider following sections: *Business Description*, *Company Background*, *Risk*, *Management* and *Legal Issues*.

8 https://www.sec.gov/edgar.shtml

To construct the final sample by applying the filters defined by L&M (Loughran and Mcdonald, 2011):

- the stock must have a match with CRSP's permanent identifier PERMNO
- the stock must be common equity
- the stock pre-filing price must be greater than $3
- the stock must have a positive book-to-market value
- CRSP's market capitalization and stock return data must be available at least 60 trading days before and after the filing date.
- firms must be traded on Nasdaq, NYSE or AMEX
- filings must contain at least 2000 words

As a result, we obtain a corpus of 60,432 10-Ks. We tokenize the corpus using NLTK[9] and lowercase it and remove punctuation.

We use word2vec CBOW with hierarchical softmax to learn word embeddings from 10-Ks. We set the dimensions of word vectors to 400, train one epoch, and use word2vec's default hyperparameters. SVMs are trained on word embeddings as described in 3.3.3. We set the confidence threshold θ to 0.8, so only words with converted SVM scores greater than 0.8 will be included in our dictionaries.[10]

As described in 3.3, we compare manually adapted and automatically adapted dictionaries (Q1) and examine whether the automatic adaptation of the L&M lexicon further improves performance (Q2). Our experimental setup is a two-way Ordinary Least Squares (OLS) robust cluster regression with dimensions of time and firm. The dependent financial variable is excess return or volatility. Independent variables are financial variables as well as one or more text variables (described in section 3.3).

To evaluate the explanatory power of sentiment dictionaries, we look at the regression coefficients that represent the strength of the association between the text variable and the dependent variable. If this association is significant, then it is unlikely that this result is due to chance. We furthermore calculate the standardized regression coefficient (the product of regression coefficient and standard deviation). This allows normalization of different value ranges of variables. It shows how the dependent variable changes if we increase a textual independent variable by a standard deviation. The standardized coefficient allows a fair comparison between text variables that have different sizes, e.g., between a text variable that has high values (many tokens per document) with one that has low values (few tokens per document).

9 https://www.nltk.org/

10 We choose this threshold because the proportion of negative, litigious and uncertain words in 10-Ks for 0.8 is roughly the same as when using L&M dictionaries.

var	coeff	std coeff	t	R^2
neg_{lm}	-0.202**	-0.080	**-2.56**	1.02
lit_{lm}	-0.0291	-0.026	-0.83	1.00
unc_{lm}	-0.215*	-0.064	**-1.91**	1.01
$H4N_{RE}$	-0.764***	*-0.229*	**-3.04**	1.05

*p ⩽ 0.05, **p ⩽ 0.01, ***p ⩽ 0.001

Table 2: Excess return regression results for L&M dictionaries and reclassified H4N dictionary. **For all tables in this chapter, significant** t **values are bolded and best standard coefficients per category are in italics** (Sedinkina, Breitkopf, and Schütze, 2019).

var	coeff	std coeff	t	R^2
$H4N_{RE}$	-0.88**	*-0.264*	**-2.19**	1.05
neg_{lm}	0.062	0.024	0.48	
$H4N_{RE}$	-0.757***	-0.227	**-2.90**	1.05
lit_{lm}	-0.351	*-0.315*	-0.013	
$H4N_{RE}$	-0.746***	*-0.223*	**-2.89**	1.05
unc_{lm}	-0.45	-0.135	-0.45	

*p ⩽ 0.05, **p ⩽ 0.01, ***p ⩽ 0.001

Table 3: Excess return regression results for multiple text variables. This table shows results for three regressions that combine $H4N_{RE}$ with each of the three L&M dictionaries (Sedinkina, Breitkopf, and Schütze, 2019).

var	coeff	std coeff	t	R^2
neg_{lm}	-0.202**	-0.080	**-2.56**	1.02
neg_{spec}	0.0102	0.0132	0.27	1.00
neg_{RE}	-0.37***	*-0.111*	**-2.96**	1.03
neg_{ADD}	-0.033	-0.0231	-1.03	1.00
neg_{RE+ADD}	-0.08**	-0.072	**-2.19**	1.03
lit_{lm}	-0.0291	-0.026	-0.83	1.00
lit_{RE}	-0.056	*-0.028*	-0.55	1.00
lit_{ADD}	-0.0195	-0.0156	-0.70	1.00
lit_{RE+ADD}	-0.0163	-0.0211	-0.69	1.00
unc_{lm}	-0.215*	-0.064	**-1.91**	1.01
unc_{RE}	-0.377***	*-0.075*	**-2.77**	1.02
unc_{ADD}	0.0217	0.0065	0.21	1.00
unc_{RE+ADD}	-0.0315	-0.0157	-0.45	1.00

*p $\leqslant$ 0.05, **p $\leqslant$ 0.01, ***p $\leqslant$ 0.001

Table 4: Excess return regression results for L&M, **RE** and **ADD** dictionaries (Sedinkina, Breitkopf, and Schütze, 2019).

3.4.1 *Excess Return*

Table 2 shows the regression results for excess return, comparing our automatically adapted dictionary ($H4N_{RE}$) with the three manually adapted L&M dictionaries (neg_{lm}, lit_{lm} and unc_{lm}). As anticipated, the coefficients have a negative sign – negative excess returns are most highly correlated with pessimistic words in 10-Ks.

For excess return, all of the L&M word lists are significant except for litigious. Especially, neg_{lm} has a big impact on the prediction of excess return: it has the highest standard coefficient (-0.080) and the highest significance (-2.56). The word list unc_{lm} performs slightly worse but is also statistically significant (-1.91). However, our adapted $H4N_{RE}$ dictionary outperforms all L&M word lists: it is highly significant (-3.04) and its standard coefficient is -0.229 which is larger by a factor of more than 2 compared to neg_{lm}. Thus, higher proportions of $H4N_{RE}$ words are especially associated with lower excess returns. This evidence supports the hypothesis that the automatically created $H4N_{RE}$ dictionary has a higher impact on excess return than the manually created L&M dictionaries. This answers our question Q1: for excess return, automatic adaptation is better than manual adaptation.

Table 3 reports *manual plus automatic* experiments with *multiple* text variables included in one regression – the combination of $H4N_{RE}$ with each of the L&M word lists. We observe that all three L&M variables

var	coeff	std coeff	t	R^2
neg_{lm}	0.118***	0.0472	**3.30**	60.1
lit_{lm}	-0.0081	-0.0073	-0.62	60.0
unc_{lm}	0.119*	0.0356	**2.25**	60.0
$H4N_{RE}$	0.577***	*0.173*	**4.40**	60.3

*p ⩽ 0.05, **p ⩽ 0.01, ***p ⩽ 0.001

Table 5: Volatility regression results for L&M dictionaries and reclassified H4N dictionary (Sedinkina, Breitkopf, and Schütze, 2019).

are not significant any more after we add $H4N_{RE}$ dictionary in regression: the explanatory power of L&M variables gets lost. On the contrary, $H4N_{RE}$ remains significant and has large standard coefficients in all experiments. More manual plus automatic experiments are illustrated in the appendix. These experiments further prove the hypothesis that automatic adaptation beats manual adaptation.

Table 4 demonstrates regression results for automatically adapting the L&M dictionaries. Experiments with multiple text variables included in one regression can also be found in the appendix. The subscript "RE+ADD" stays for a dictionary that combines RE and ADD; e.g., unc_{RE+ADD} is the union of unc_{RE} and unc_{ADD}.

Each category (neg, lit and unc) of the automatically adapted dictionary outperforms original manually adapted word lists; e.g., the standard coefficient of neg_{RE} (-0.111) is better than that of neg_{lm} (-0.080). The same observation applies to the unc category – the standard coefficient of unc_{RE} (-1.91) is better than unc_{lm} (-2.77). Results are statistically significant for neg_{RE} (-2.96) and unc_{RE} (-2.77). We also perform experiments with neg_{spec}, the negative word list of Hamilton et al. (2016). neg_{spec} does not provide good results: it is not significant.

These experimental results answer our question Q2: automatic adaptation of L&M's manually adapted dictionaries further improves their performance.

3.4.2 *Volatility*

Table 5 gives regression results for volatility, comparing $H4N_{RE}$ and L&M word lists. Except for litigious, the coefficients have positive signs, so the more pessimistic words (as measured by the Harvard or Fin-Neg word lists) that appear in the 10-K, the higher is the volatility.

Results for neg_{lm}, unc_{lm} and $H4N_{RE}$ indicate that these word lists have a significant impact on volatility. The dictionary neg_{lm} does a better job at explaining volatility than other L&M word lists ($t = 3.30$). However, the best performing dictionary is again automatically adapted $H4N_{RE}$. It has the standard coefficient of 0.173 which

var	coeff	std coeff	t	R^2
$H4N_{RE}$	0.748***	*0.224*	**4.44**	1.11
neg_{lm}	-0.096*	-0.038	-2.55	
$H4N_{RE}$	0.642***	*0.192*	**4.28**	1.11
lit_{lm}	-0.041*	-0.037	-2.54	
$H4N_{RE}$	0.695***	0.208	**4.54**	1.11
unc_{lm}	-0.931**	*-0.279*	-2.73	

*p $\leqslant$ 0.05, **p $\leqslant$ 0.01, ***p $\leqslant$ 0.001

Table 6: Volatility regression results for multiple text variables (Sedinkina, Breitkopf, and Schütze, 2019).

is three times larger than that of neg_{lm} (0.0472). This highlights that $H4N_{RE}$ has a higher explanatory value than the L&M dictionaries and thereby provides an answer to question Q1 – automatic adaptation performs better than manual adaptation. Table 6 also verifies this: manual plus automatic experiments of $H4N_{RE}$ with one of the L&M dictionaries demonstrate statistically significant results for $H4N_{RE}$. On the contrary, L&M dictionaries become negative in sign, indicating that more negative, uncertain, or litigious words decrease volatility. It means that L&M words are not effective in measuring tone in 10-Ks, i.e., there is no correlation between volatility and negative tone in 10-Ks in this regression setup. Additional manual plus automatic experiments illustrate the same observation (see the appendix).

Table 7 summarizes regression results for automatically adapting the L&M dictionaries. Experiments with multiple text variables in one regression can be found in the appendix. neg_{RE} outperforms neg_{lm} dictionary – its standard coefficient (0.0657) is better by about 40% than that of neg_{lm} (0.0472). The results for a negative word list of (Hamilton et al., 2016), neg_{spec}, are not significant and negatively signed, meaning that an increase of pessimistic words decreases volatility. For lit, neither L&M nor adapted dictionaries are significant. The unc_{RE} dictionary performs only slightly worse than unc_{lm} (0.0344 vs. 0.0356 for the standard coefficients). neg_{RE} provides the best results – its standard coefficient is 0.0657 and $t = 3.57$.

Regression results for volatility indicate that negative words perform better than uncertain words, even though L&M designed the unc_{lm} dictionary specifically for volatility. This applies for both: for L&M dictionaries (neg_{lm}) and their automatic adaptations (e.g., neg_{RE}).

var	coeff	std coeff	t	R^2
neg_{lm}	0.118***	0.0472	**3.30**	60.1
neg_{spec}	-0.038	-0.0494	-2.73	60.1
neg_{RE}	0.219***	*0.0657*	**3.57**	60.1
neg_{ADD}	0.032***	0.0224	**4.06**	60.0
neg_{RE+ADD}	0.038***	0.0342	**4.32**	60.1
lit_{lm}	-0.0081	-0.0073	-0.62	60.0
lit_{RE}	0.0080	0.0040	0.20	60.0
lit_{ADD}	0.028	*0.0224*	1.07	60.0
lit_{RE+ADD}	0.015	0.0195	0.81	60.0
unc_{lm}	0.119*	*0.0356*	**2.25**	60.0
unc_{spec}	-0.043	-0.0344	-1.56	60.0
unc_{RE}	0.167*	0.0334	**2.30**	60.0
unc_{ADD}	-0.013	-0.0039	-0.17	60.0
unc_{RE+ADD}	0.035	0.0175	0.68	60.0

*p $\leqslant$ 0.05, **p $\leqslant$ 0.01, ***p $\leqslant$ 0.001

Table 7: Volatility regression results for L&M, **RE** and **ADD** dictionaries (Sedinkina, Breitkopf, and Schütze, 2019).

Table 7 also shows results for unc_{spec}, the uncertainty dictionary of Theil, Stajner, and Stuckenschmidt (2018). It does not perform well: it is not significant and the coefficient has the positive sign.[11]

neg_{RE} performs better than neg_{lm}. Hence, the best automatic adaptation of an L&M dictionary has a more explanatory value than the best L&M dictionary. This finding supported by Table 7 confirms our answer to Q2: automatic domain adaptation can further improve manual adaptation.

3.5 ANALYSIS AND DISCUSSION

In this section, we perform a qualitative analysis of the differences between the classifications of L&M and those of our sentiment dictionaries. This can help to understand the reasons for the discrepancy in performance.

Table 8 lists example words from our automatically adapted dictionaries. The **ADD** words are those that L&M classified as nonimportant for a category. So words such as "diminishment" (neg), "es-

11 Theil, Stajner, and Stuckenschmidt (2018) specify volatility for the period [6 28] whereas we use days [6 252], based on (Loughran and Mcdonald, 2011). Larger time windows allow more reliable results and take into account the fact that information disclosures can impact volatility for long periods (Belo et al., 2016).

ADD_{neg}	missing, diminishment, disabling, overuse
ADD_{unc}	reevaluate, swings, expectation, estimate
ADD_{lit}	lender, assignors, trustee, insurers
RE_{neg}	confusion, unlawful, convicted, breach
RE_{unc}	variability, fluctuation, variations, variation
RE_{lit}	courts, crossclaim, conciliation, abeyance
$H4N_{RE}$	compromise, issues, problems, impair, hurt

Table 8: Word classification examples from automatically adapted dictionaries (Sedinkina, Breitkopf, and Schütze, 2019).

timate" (unc) and "trustee" (lit) were classified as relevant words by our system and appear to imply negativity, uncertainty and litigiousness, respectively, in financial domain.

According to L&M's classification scheme, a word can belong to several categories, e.g., L&M classify "unlawful", "convicted" and "breach" both as negative and as litigious. Our RE method labels these words only as negative, not as litigious. Correspondingly, L&M classify "confusion" as negative and uncertain, but automatic RE adaptation labels it only negative. This illustrates that there is strong distributional evidence in the 10-Ks for negativity, but weaker distributional evidence for the categories litigious and uncertain. In our work, only "negative" uncertain/litigious words have a high impact on financial outcomes. In contrast, positive words like "acquittal" (positive litigious) and "suspense" (positive uncertain) do not help in predicting financial variables. This explains the reason why the negative category provides better results in our adaptation than the other two.

An interesting example of a discrepancy in classification for RE is the word "abeyance". This word has a domain-general uncertain meaning – "something that is waiting to be acted upon". So, L&M classify it as uncertain. However, in 10-Ks "abeyance" is mostly used in legal contexts: "held in abeyance", "appeal in abeyance". As a result, our automatic adaptation classifies it as litigious. The nearest neighbors of this word in embedding space are also litigious words: "stayed", "hearings" and "mediation".

Our $H4N_{RE}$ dictionary includes 74 "common" words from the H4N word list. For example, these words are "compromise", "serious" and "god". The word "compromise" has a negative meaning in 10-Ks: its nearest neighbors in the embedding space include negative words like "misappropriate", "breaches", "jeopardize". In a general-domain embedding space,[12] the nearest neighbors of "compromise" are not negative, i.e., "negotiated settlement", "accord" and "modus vivendi".

12 https://code.google.com/archive/p/word2vec/

	neg_{lm}	lit_{lm}	unc_{lm}	neg_{ADD}	lit_{ADD}	unc_{ADD}	neg_{RE}	lit_{RE}	unc_{RE}	$H4N_{neg}$	$H4N_{cmn}$	$H4N_{RE}$
neg_{lm}		7	2	0	0	0	49	2	0	48	52	12
lit_{lm}	17		0	0	0	0	6	20	0	7	93	1
unc_{lm}	14	0		0	0	0	18	2	30	16	84	2
neg_{ADD}	0	0	0		0	0	0	0	0	18	82	2
lit_{ADD}	0	0	0	0		0	0	0	0	1	99	0
unc_{ADD}	0	0	0	0	0		0	0	0	3	97	0
neg_{RE}	95	5	4	0	0	0		0	1	52	48	21
lit_{RE}	18	86	2	0	0	0	0		0	7	93	0
unc_{RE}	11	2	92	0	0	0	10	0		13	87	3
$H4N_{neg}$	27	2	1	10	0	0	15	0	0		0	6
$H4N_{cmn}$	2	1	0	2	1	0	1	0	0	0		0
$H4N_{RE}$	79	2	2	17	0	0	74	0	1	78	22	

Table 9: Quantitative analysis of dictionaries. For a row dictionary d_r and a column dictionary d_c, a cell gives $|d_r \cap d_c|/|d_r|$ as a percentage. cmn = common (Sedinkina, Breitkopf, and Schütze, 2019).

It means that the word "compromise" is used in positive contexts in the general domain. Hence, this underlines the importance of domain-specific word embeddings because they allow capturing domain-specific information.

Another example of a negative category is the word "god"; it is mostly used in 10-Ks in the phrase "act of God", indicating that the company cannot fulfill obligations due to unforeseen and unavoidable occurrences. Its nearest neighbors in the 10-K embedding space are "terrorism" and "war", showing that this word has a negative sense in 10-Ks. In general-domain, this word is labeled as positive. Thus, the annotation mistakes are likely to be made when the context is not present to the annotators. So, when the annotators consider the word in the context ("act of God"), the meaning of this word can be recognized and thereby annotated correctly.

This analysis has led us to conclude that manual annotation of words without context is error-prone. It should be performed based on the word's contexts in the target domain (like our automatic adaptation). This will allow avoiding the errors in the annotation because it will be based not on the prior belief about word meanings but the word's contexts.

3.5.1 *Quantitative Analysis*

Table 9 presents a quantitative analysis of words in the dictionaries. Each cell in the dictionary is the percentage $|d_r \cap d_c|/|d_r|$ where d_r is a dictionary and d_c is a column. Diagonal entries are omitted since they are all equal to 100%. The table gives us an intuition about the relationship between different dictionaries and also between the categories negative, litigious, and uncertain. For example, row "neg_{lm}", column "neg_{RE}" means that 49% of the words in neg_{lm} are also members of neg_{RE}.

Considering rows neg_{lm}, lit_{lm} and unc_{lm}, we can understand how L&M constructed their dictionaries. Negative word list neg_{lm} consists of words as well from $H4N_{neg}$ as from $H4N_{cmn}$, namely in almost equal proportions. Many "common" words from the H4N dictionary were classified as negative by L&M for the financial domain. Most lit_{lm} and unc_{lm} words are part of $H4N_{cmn}$ dictionary, relatively few are from $H4N_{neg}$. Surprisingly, only 12% of neg_{lm} words were automatically identified as negative in domain adaptation and added to $H4N_{RE}$. Since $H4N_{RE}$ performs better than neg_{lm} in our experiments, this shows that the annotators can misclassify the words in the financial domain if the actual contexts of the words are not considered.

We see two types of errors in the human annotation. First, as discussed in 3.5, words like "god" are labeled incorrectly because the widely used context in 10-Ks ("act of God") is not explicitly present to the annotator. Second, the words often express the strength of the sentiment, not only the sentiment itself. So, many neg_{lm} words are just slightly negative and do not contribute to explaining financial outcomes like excess return or volatility. Our automatic adaptation method classifies them as neutral. The strength of sentiment of a word is not easy to estimate by human annotators. If we look at the row $H4N_{RE}$, we observe that most of its words are part of neg_{lm} dictionary, namely 79%, and only a few words are from lit_{lm} and unc_{lm} (2% each). This statistic can be interpreted as indicating that L&M had a high recall (they find the most of the impactful words), but low precision (only 12% of their negative words are added to the $H4N_{RE}$). $H4N_{RE}$ dictionary consists of 78% of $H4N_{neg}$ words and 22% of $H4N_{cmn}$ words. This confirms the fact that many common words can be negative in financial contexts. So, it is necessary to perform domain adaptation.

We finally look at the distribution of negative, litigious and uncertain categories in the L&M, ADD, and RE dictionaries, namely how they overlap. We see that lit_{lm} and unc_{lm} are in neg_{lm} dictionary (17% and 14%), while they do not overlap with each other. Negative, litigious and uncertain categories of ADD dictionaries (neg_{ADD}, lit_{ADD}, unc_{ADD}) do not have the overlap at all. As concerns the RE dictionary, there is no overlap between RE dictionaries except for unc_{RE}

and neg_{RE} – 10% of the words of unc_{RE} are part of neg_{RE}. Hence, we see that L&M dictionaries have a higher overlap than ADD and RE dictionaries. So, automatic domain adaptation allows us to more clearly distinguish between the three categories – negative, litigious, and uncertain – than manual L&M domain adaptation.

3.6 CONCLUSION

We automatically created sentiment dictionaries for predicting financial outcomes excess return and volatility. Our experiments demonstrated that the automatically adapted sentiment dictionaries outperform the manually adapted dictionaries. In particular, we achieve a new state of the art in predicting financial outcomes excess return and volatility.

Our quantitative and qualitative study allowed us to obtain insights into the semantics of the dictionaries. We revealed that manual adaptation of dictionaries is error-prone due to *the expert's a priori belief* about a word's meaning. Instead, the annotation should be performed based on the word's *contexts in the target domain*.

This project is an object of a long-term perspective. For example, we can explore whether the word embeddings trained on much larger corpora can further be beneficial for domain adaptation. This suggests the following direction for future research: the comparison of domain adaptation based on our domain-specific word embeddings vs. based on word embeddings trained on much larger corpora. Another interesting topic for future research is to investigate how the meaning of the word changes over time and whether these changes in the meaning can significantly impact the prediction.

4

TEMPORAL TRANSFER LEARNING WITH PRIORS FOR FINANCIAL PREDICTION FROM TEXT

4.1 INTRODUCTION

The growing volume of financial reports, news articles, and social media discussions has advanced the field of *natural language based financial forecasting*. Social media has especially a tremendous impact on business. For example, IBM and Amazon demonstrate that online blog postings can successfully predict spikes in the sales rank of books (Gruhl et al., 2005). Other researchers predict future sales by exploring social media forums such as Twitter and Weblogs (Liu et al., 2007). Reddit[1] is another example of such public online discussion forums with a wide variety of subreddits, each discussing a particular subject. Reddit is therefore a good data source to understand and analyze the feedback of millions of potential consumers. In our conceptual framework, we use the discussions from *movie subreddits* to predict future movie sales. Furthermore, we consider in our work another important factor for financial forecasting – the temporal dynamics. The language of social media is changing very rapidly. Every day social media provides a wide range of opinions and commentary about many topics and trends. As a result, new words appear in the vocabulary, new named entities gain sudden importance and new trends are rapidly emerging. To our best knowledge, this is the first work that uses Reddit data for financial forecasting under the consideration of language change over time.

Finance methods usually operate with *time series data*, i.e., data gathered sequentially in time. A classic technique to perform a time series analysis are autoregressive models (Liu et al., 2016; Moniz and Jong, 2014; Nofer and Hinz, 2015; Wang, Huang, and Wang, 2012). In our work, we apply a different time-dependence approach and rely only on textual information. We use Linear Regression (*LR*) and continually base the newly inferred parameters on old parameters and new data. We formulate the task as *temporal transfer-learning*, where the test samples consist of future observations relative to the training samples. So, the models are trained incrementally by the continuous update.

In specific domains like finance, more training data is not always helpful. This is because changes in the macroeconomy and specific businesses make older reports less relevant for prediction (Kogan et al., 2009). Temporal transfer-learning solves the problem of training data selection as we do not use it all at once. We consider all training

1 http://www.reddit.com

samples by continuously moving a window over them. So, this allows us to leverage the information contained in the training signals of previous models and at the same time to pay attention to those features that matter.

In our work, we propose, therefore, a model that applies transfer learning in a time-dependent manner by using a parameter prior based on previous models for estimation at the current time-steps. This approach is applied iteratively so that the prior is adapted continuously in a Bayesian manner. We use the LR algorithm and a Gaussian prior on the parameters (which can be implemented easily similar to l2-regularization). We show that **temporal transfer-learning** improves performance when compared to the method that only uses the recent data for training. We also demonstrate the benefit of our approach over the autoregressive models. Moreover, our method outperforms a model that "naively" uses all training data at once. Unlike the "naive" method, our approach allows updating already existing models with new information, without the necessity to train a model again from scratch.

4.2 RELATED WORK

Temporal transfer learning methods were studied in the literature under various terms. Among these are the algorithms that handle the problem of *concept drift* when the relation between the input data and the target variable changes over time. To react to concept drifts, *adaptive learning* has been proposed to update predictive models online (Gama et al., 2014). This approach is based on feedback that is used to update the model. In case the feedback is not available, the algorithm estimates the ground truth approximately by retrospectively inspecting the historical data. This technique was applied in multiple domains, for example, monitoring and control (e.g., identification of anomalous behavior in the web), management and strategic planning (e.g., evaluation of creditworthiness), personal assistance and information (e.g., customer profiling for marketing), and ubiquitous environment applications (e.g., mobile vehicles).

In the field of natural language processing, online algorithms have been developed to handle large datasets (Dredze and Crammer, 2008; Hoffman, Bach, and Blei, 2010). For example, the online algorithm of Hoffman, Bach, and Blei (2010) allows improving topic modeling on large datasets including those arriving in a stream. The online algorithm of Dredze and Crammer (2008) proposes a multi-domain learning framework that combines the parameters from multiple classifiers to adapt multiple source domain classifiers to a new target domain. Temporal transfer learning was also used in the medical domain, for example, to predict and prevent cardiac arrest (Ho and Park, 2017). The authors claim that the use of information from adjacent

time points allows to overcome small sample size issues and capture temporal trends in the data. Our method applies a similar approach, however, it solely relies on the textual information and addresses the field of finance.

A classic method to perform financial forecasting is a time series analysis. For instance, autoregressive (*AR*) models predict the financial performance by using a linear combination of the observations in the previous days. In vector autoregression (*VAR*), the model considers, in addition to the observations of the previous days, textual information as observed in the previous days (Nofer and Hinz, 2015). Some research adopts time series models like autoregressive integrated moving average (ARIMA) or generalized autoregressive conditional heteroskedasticity (GARCH) and combines them with machine learning techniques (Liu et al., 2016; Moniz and Jong, 2014). Many studies also attempted to build financial forecasting systems based on various NLP techniques. Researchers exploited different text resources to analyze how the textual aspects can affect the market, e.g., price (Kazemian, Zhao, and Penn, 2016; Wang et al., 2013), volatility (Rekabsaz et al., 2017), and potential risks (Nopp and Hanbury, 2015). Other studies investigated whether the textual information from public discussion forums can impact customer behavior and therefore be useful for predicting sales performance (Gruhl et al., 2005; Liu et al., 2007). Some systems employ bag-of-words NLP techniques (Kogan et al., 2009). Others use sentiment analysis (Liu et al., 2007; Si et al., 2013). Apart from real-world quantity forecasting (Kogan et al., 2009), several studies view financial prediction as a classification problem, e.g., by predicting "up" and "down" price trends (Lee et al., 2014). Our work differs from existing research in two important respects. First, we want to enable our model to use textual information by **learning continuously without forgetting**. Second, by using parameters prior based on previous models, we help the model **focus its attention on those features that matter** as previous models provide additional evidence for the relevance or irrelevance of those features. We, therefore, develop a real-world application scenario where a new model becomes more complex by leveraging its experience.

4.3 MOVIE SALES PREDICTION

To predict movie sales, we use US movies box office weekend returns[2], i.e., for each movie, we obtain the information about the gross income of that movie on the particular weekend. We consider the gross incomes between 2009 and 2018, with the last two years (2017 –

2 We focus on the weekend data rather than daily incomes since this normalizes gross incomes on different dates, preventing movies from having a higher income just because they were shown on the weekend.

2018) reserved for the final evaluation, the previous two (2015 – 2016) for development, and the remainder for training.

On Reddit, people initiate a discussion thread with a post every day, and others respond with comments. In our work, we use both posts and comments of two popular subreddits *movies* and *entertainment*. We index these subreddits using *Elasticsearch*[3]. Then, for each movie from the US movies box office, we collect all relevant posts and comments appearing in our Reddit index. A post was considered "relevant" to a movie if the following conditions hold:

- The post contained the exact movie name that appeared in the post.
- The date of the post is within a window starting on Monday and ending on Sunday (the week before the movie's weekend income date).

Table 10 gives statistical information about the resulting dataset.

Dataset	# of posts	# of words	# of movies
Train	12,249	398,702	754
Dev	5,772	158,380	296
Test	5,885	210,505	299

Table 10: Reddit datasets discussing the movies.

We align discussions (during a week) about movies with income changes for these movies (at the end of a week). The income change of a weekend w is defined as[4]:

$$\log \frac{\text{gross}_w}{\text{gross}_{w-1}} \tag{9}$$

where gross_w is the gross income of the weekend w and gross_{w-1} is the gross income of the previous weekend. Table 11 shows an example of the gross income development for the movie "Coco" and the number of opinion words that appeared in posts of the previous 7 days (Monday till Sunday). We use Hu and Liu (2004) opinion lexicon[5] that contains positive, negative, and neutral opinion words for English language (in total 6866 words). The rationale behind this is that sentiment-oriented words, such as "good" or "bad", are more indicative than other words (Zhang and Varadarajan, 2006). Gross_w was furthermore normalized to produce an "Income per Screen." For the normalization, we used the number of theaters the movie was shown on. This allows comparing sales of blockbuster movies, sometimes released to 4000 screens, to lower-profile movies released to 1000-2000 screens.

3 https://www.elastic.co/

4 Note that we keep only those weekends for which we find relevant posts.

5 https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

Liu et al. (2007) demonstrated that people's opinions (e.g., reflected by the post's sentiment) can be a good indicator of how the box office performance evolves. Gruhl et al. (2005) also showed a correlation between the number of postings and the sales rank of the product. We also assume that the number of discussions about the movie may impact the gross income. As an example, Figure 4 shows a relationship between the volume of discussion during a week and gross income on the weekend. The upper plot shows the change of box office revenues of the movie "Coco", and the lower plot shows the change in the number of opinion words discussing this movie. A spike in the number of words for the weekend Feb. 2 – 4 indicates that a large volume of discussions on that movie appeared during this week (Jan. 29 – Feb. 4). The change in box office revenue for this weekend is also high and positive. It means that the volume of discussions may have a positive effect on movie sales. On the contrary, sales can be negatively affected by little discussions about the movie. The Figure 4 confirms this fact, showing that the decrease in sales is correlated with the lower volume of discussions (e.g. for the weekend Feb. 9 – 11)[6].

Year	Weekend	Change	# of opinion words
2018	Jan. 5-8	-0.278	3
2018	Jan. 12-14	-0.381	3
2018	Jan. 19-21	-0.427	9
2018	Jan. 26-28	-0.249	13
2018	Feb. 2-4	0.205	18
2018	Feb. 9-11	-0.488	5
2018	Feb. 16-18	-0.222	10
2018	Feb. 23-25	-0.319	11

Table 11: Weekend US movie box office returns for the movie "Coco" and the number of opinion words about this movie during that week.

Below are Reddit snippets for the movie "Coco" on different dates. Opinion words are bolded.

Feb 2 – 4: *"We **just** got Coco **so** it'll be a while until they pander to us again. **Though admittedly**, Coco was fucking **amazing**.", "Coco was one of the **best animated** films I have seen in a **long** time!","I've seen the Breadwinner. It's **pretty good**, but Coco is **better**. It's on the whole a **pretty weak** year for animation **anyway**."*

6 Note that that the number of opinion words discussing the movie may not be an accurate indicator of a product's sales performance. A product can attract much attention (thus a large number of posts) due to various reasons such as aggressive marketing, unique features, or being controversial (Liu et al., 2007).

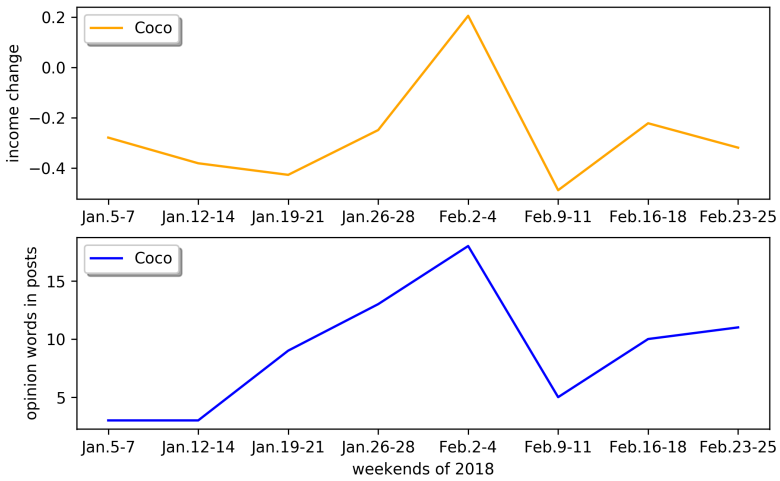

Figure 4: The lower plot is the number of opinion words discussing the movie "'Coco". The upper plot is the change of box office revenues over weekends. We see a relationship between the number of opinion words and the change in gross income – the increase of discussions increases future gross income.

Feb 9 – 11: *"I **love** coco!", "Coco was fucking **delightful**.", "You missed Coco at number 12. In the **top** 20 Coco, Dunkirk, Get Out, and Boss Baby are all **original** ideas no?",*

Although a second example snippet has a positive sentiment, gross income on Feb. 9 – 11 is lower than on Feb. 2 – 4 (see Figure 4). This movie is, however, more often discussed on Feb. 2 – 4. So, this shows that the volume of discussions is a more important predictor of future financial performance in this case. Nevertheless, a big volume of negative or positive discussions may still impact the gross income, respectively negatively or positively (Liu et al., 2007). Thus, two important components must be considered in the forecasting – the number of discussions and its sentiment. That is why we use not all the words appearing in the posts, but only a set of opinion words (negative, positive and neutral), and use their counts in a post as a feature vector.

4.4 TRAINING PROTOCOL

Given a collection of data points (movies) X and corresponding target values y (change in gross income), we wish to find the weight vector w that approximately associates data points x_i with their corresponding labels y_i at time t:

$$\hat{y}^{(t)} = X^{(t)} w^{(t)} + b^{(t)} \tag{10}$$

The training objective for minimization at a time interval t is the mean squared error over all examples:

$$l(y^{(t)}, \hat{y}^{(t)}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}^{(t)} - y_i^{(t)})^2 + R(\mathbf{w}^{(t)}; \mathbf{w}^{(t-1)}) \tag{11}$$

$\mathbf{X}^{(t)}$ contains the bag of word vectors of 6866 opinion words for all movies in the period t, and $\mathbf{y}^{(t)}$ contains the target values (the changes in gross income). $R(\mathbf{w}^{(t)}; \mathbf{w}^{(t-1)})$ is the regularization term that also models time-dependence, as discussed below.

The time unit is a week, i.e. the Reddit posts discussing the movie during the week. We wish to predict the change in the gross income at the end of this week. We construct the model at first from the period t, then we move the window and build a new model from the period $t+1$. The new model considers information from the period t as it uses parameters from this model. Thus, in each step, we continuously infer the parameters so that in the last step we implicitly incorporate the information from all previous models. The knowledge transfer is accomplished by using the weight values from the previous model as the mean of a Gaussian prior for the weights in the new model. The Gaussian prior is equivalent to an l2-regularization term where the variance is incorporated into the regularization strength (Murphy, 2012). We combine the prior information with a standard l2-regularization and obtain the following regularization term:

$$R(\mathbf{w}^{(t)}; \mathbf{w}^{(t-1)}) = \alpha \cdot ||\mathbf{w}^{(t)}||_2^2 + \beta \cdot ||\mathbf{w}^{(t)} - \mathbf{w}^{(t-1)}||_2^2$$

where $\mathbf{w}^{(t)}$ are the parameters optimized in iteration t, and α and β control regularization strength. Here, if β is zero then the model relies only on the current data $\mathbf{y}^{(t)}, \mathbf{X}^{(t)}$. If β is very large then it will tend to ignore current parameters and will mainly use the weight values $\mathbf{w}^{(t-1)}$ from previous model. We optimized the parameter α for the baseline model (trying the values $10.0, 1.0, 0.1, 0.01, 0.001$), and found the optimal value to be 0.01. We fix that value and optimize the time dependence term β over the values $10.0, 1.0, 0.1, 0.01, 0.001$, which resulted in $\beta = 0.01$. For the optimization, we utilize the development data, i.e. Reddit posts submitted between 2015 and 2016.

We apply mini-batch gradient descent for solving the optimization problem setting the batch size to 128. To train the model, we use 100 epochs and a learning rate of 0.01. Gradients for backpropagation were estimated using the Adam optimizer (Kingma and Ba, 2015)

4.5 EXPERIMENTS

We use weekend data between 2009 and 2014 for training. For development, we use the data of 2015 and 2016 and test the model in 2017-2018.

4.5.1 *Temporal Transfer Learning ($\beta = 0.01$)*

To predict the gross income change of a particular weekend w_i, we use training data of the last 7 days before this weekend. Setting β to 0.01, we continually transfer the parameters from the prior models $w_0 ... w_{i-1}$. For example, we wish to predict the gross income change for the weekend Jan. 15-17 of 2009. For this, we train the model using Reddit posts that appeared in the period from 11 Jan. (Monday) to 17 Jan. (Sunday) of 2009 and the parameters from the prior model that was trained on the data of Jan. 8-10 of 2009. In the next step, we predict the changing income for the following weekend (Jan. 22-24) using the data of that week and the parameters of the prior model (Jan. 15-17). In this way, we obtain the predictions for all weekends until 2014[7]. Overall, we construct 295 models for the period from 2009 to 2014.

To evaluate the models, we apply the same procedure. So, to predict the change in gross income for the first weekend of 2015, we use the model of the last weekend of 2014 (with implicit information from all previous models). To predict the change for the 2d weekend of 2015, we construct a new model using the data of the first weekend of 2015 and parameters from the model of the last weekend of 2014. Thus, the last model (the last weekend of 2018) contains implicit information from all prior models.

4.5.2 *Regression Baseline 1 ($\beta = 0$)*

We compare our method to several baselines. To predict the gross income change of a particular weekend w_i, we only use the training data of the previous weekend. We can achieve this by setting β to 0. In this way, we ignore the parameters from the prior models $w_0 ... w_{i-1}$.

4.5.3 *Regression Baseline 2 (all data)*

In this baseline, we use all available data for training at once. For example, we use the whole data from 2009 to 2014 for training and then test the model on the first weekend of 2015. For testing on the 2d weekend of 2015, we add to the training data for the first weekend of 2015. We follow this technique until we predict the last weekend of 2018. This testing scheme guarantees a fair comparison with the temporal transfer learning method as the development and test data is also used for training. Each time we want to predict the income change for a particular weekend, we construct a new model using all previous data for training at once.

7 Note that we do not use for training the weekends for which no Reddit posts were available.

4.5.4 *Time Series Baseline*

We also compare temporal transfer-learning to the time series baseline using the Auto-Regressive Integrated Moving Average (ARIMA) model (Hamilton, 1994). The name ARIMA can be understood as the combination of three elements, the AR-term, the I-term and the MA-term. The ARIMA model originates from the autoregressive model (AR-term) and the moving average model (MA-term). AR(k) assumes that each observation at time t, X_t, is a linear combination of the previous k observations. So, AR(k) satisfies $X_t = \sum_{i=1}^{k} \alpha_i X_{t-1} + \epsilon_t$ where α_i is a coefficient and ϵ_t denotes the constant term which is the average period-to-period change. This is similar to traditional multiple regression model, but X_t is regressed on past values of X_t (Liu et al., 2016). MA(q) stands for the lags of the forecast errors q: $X_t = \sum_{i=1}^{q} \beta_i \epsilon_{t-1} + \epsilon_t$ where β_i is a coefficient. I-term(d) (stands for "integrated") makes the time series stationary, e.g., by subtracting from an observation X_t an observation at the previous time step X_{t-1}: $\nabla^{d=1} X_t = X_t - X_{t-1}$ where d is the first order differences of X_t in this case. The second order differences of X_t is defined by $\nabla^{d=2} X_t = \nabla X_t - \nabla X_{t-1}$. This differencing technique eliminates the influences of trend components of data before the ARIMA model can be fitted. Thus, if the sequence of ∇X_t satisfies an ARMA(k, q), the sequence of X_t satisfies the ARIMA(k, d, q)

$$\nabla^d X_t = \sum_{i=1}^{q} \beta_i \epsilon_{t-1} + \sum_{i=1}^{k} \alpha_i \nabla^d X_{t-1} + \epsilon_t \tag{12}$$

which are parameterized by three terms k, d, q and weights vector $\alpha \in \mathbb{R}_k$ and $\beta \in \mathbb{R}_q$ (Liu et al., 2016).

ARMA(k, q) is a special case of the ARIMA(k, d, q), where the order of the differences is zero. Since we observe the change in gross income and not the gross income itself, our data is already stationary. It does not need to be stationarized through differencing (we set parameter d to 0). To decide which k and q parameters to use for our data, we use *autocorrelation function plot (ACF)* . ACF shows the correlation between the points, i.e., how a time series is correlated with its past values. Figure 5 illustrates an example of autocorrelation plot for the movie "Coco". If the time series is non-random then the autocorrelations are significantly non-zero. The horizontal lines displayed in the plot correspond to confidence bands. The dashed line is a 99% confidence band. The straight line is a 95% confidence band. We see that the autocorrelations for this movie are near zero, meaning that time series is random. We make the same observation for all other movies. A possible reason for this is the fact that our data has a limited-time series history, i.e., one movie is shown on screens only for two/three months. Thus, we set the parameters k and q to 0 meaning that we

only use the constant term ϵ_t, i.e., we apply a *random walk model* that uses the average period-to-period change (the long-term drift):

$$\hat{X}_j^{(t)} = \epsilon_t = \frac{1}{n}\sum_{i=1}^{n} X_{t-i} \tag{13}$$

where $\hat{X}_j^{(t)}$ is the predicted change in the gross income for a movie j at a time t, and X_{t-i} are previous observations for a movie j. In our work, we predict the change in gross income for each movie X_j. So, we run j ARIMA regressions at each time t. Therefore, we obtain the final equation as follows:

$$\hat{X}^{(t)} = \frac{1}{n}\sum_{j=1}^{n} \hat{X}_j^{(t)} \tag{14}$$

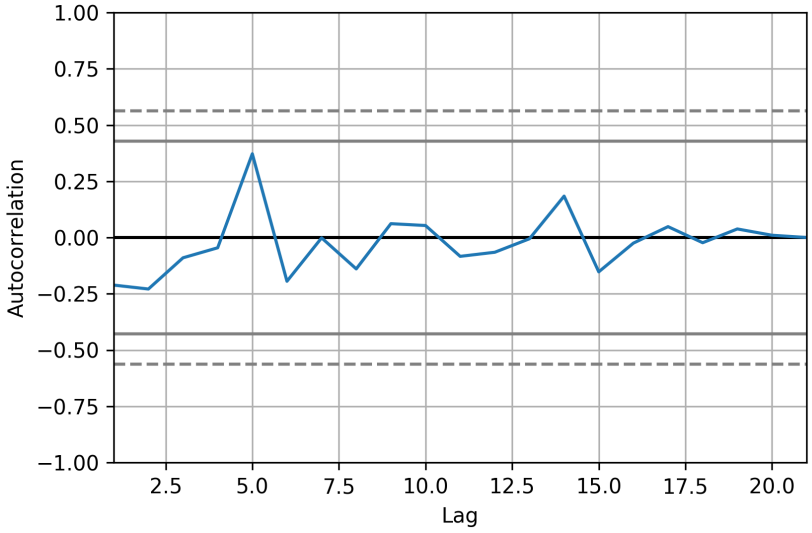

Figure 5: Autocorrelation plot for the movie "Coco". If time series is non-random then the autocorrelations are significantly non-zero. The horizontal lines correspond to confidence bands. The dashed line is 99% confidence band. The straight line is 95% confidence band. We see that the autocorrelations for the movie "'Coco" are near zero, i.e., this time series is random. Thus, we apply a *random walk model*.

4.6 RESULTS

Tables 12 shows evaluation results of different methods. As an evaluation measure, we use the average mean square error (MSE) over the weekends, i.e., over the weekends between 2015 – 2016 (development set), and 2017 – 2018 (test set).

In the first set of experiments, we compare our method with baseline 1 ($\beta = 0$). We achieve significantly better results – the improvement for development (0.56 vs. 0.35) and test (0.66 vs. 0.39) data sets is highly significant ($p \leqslant 0.001$). This shows that transferring the parameters from prior models is more beneficial than just relying upon the model from the current week.

In the second set of experiments, we compare our model with baseline 2 (all data). The difference in the performance of the development set is not significant (0.37 vs. 0.35). However, for test set, temporal transfer learning significantly outperforms the baseline 2 (0.42 vs. 0.39, $p \leqslant 0.01$). This demonstrates the benefit of our model over the model that uses previous knowledge all at once.

The last set of experiments compares our method with the time series model, ARIMA. It can be seen that MSE has substantially reduced with our model – results are statistically significant for the development and test sets ($p \leqslant 0.001$). This demonstrates that textual data plays an important role in forecasting. Furthermore, if the number of periods in time series is not large, e.g., one movie is shown on screens only during two-three months, it is appropriate to use alternative forecasting methods. Temporal transfer learning can be a reasonable choice in this case.

method	**MSE dev**	**MSE test**
baseline 1 ($\beta = 0$)	0.56	0.66
temporal transfer-learning	0.35***	0.39***
baseline 2 (all data)	0.37	0.42
temporal transfer-learning ($\beta = 0.01$)	0.35	0.39**
ARIMA	0.60	0.64
temporal transfer-learning ($\beta = 0.01$)	0.35***	0.39***

*$p \leqslant 0.05$, **$p \leqslant 0.01$, ***$p \leqslant 0.001$

Table 12: Comparison of temporal regression to the baselines: average MSE of all models (on development and test data sets).

Figure 6 is an example of the performance results of different methods. For clarity, we only plot the results of the last 14 weekends of 2018. In this Figure, we can see how different methods perform in each time step (i.e., each weekend). The plot confirms the findings from Table 12 – parameter transfer (orange color vs. blue color) is beneficial for the forecasting. In almost each time step, the MSE of our method is lower than that of the model with $\beta = 0$. When the size of the recent week's training data is too small (e.g., this is a case for the 11. weekend), our approach is especially helpful. Comparing ARIMA (red color) and temporal transfer learning (orange color) plots, we see a clear difference in the performance – we obtain better results. Figure 6 also shows that our model achieves competitive results when

compared to baseline 2 (all data). Some weekends (e.g., 10, 11) benefit from more data (green color) while other weekends (e.g., 7, 8) benefit more from temporal transfer learning (orange color). It means that in some cases more data does not necessarily improve the results. On the contrary, the performance decreases in some cases (e.g., for the 7. and 8. weekend). Temporal transfer learning in its turn shows that it is possible to update the already existing model without the performance drop when compared to the model that is trained from scratch with all data.

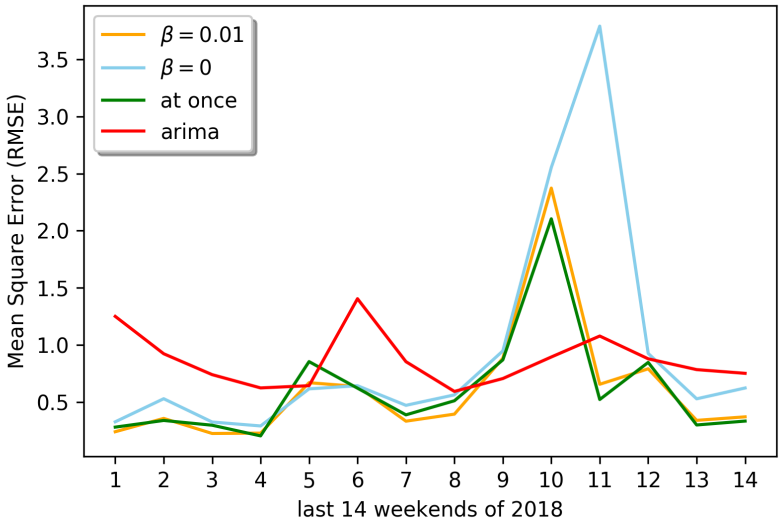

Figure 6: Illustration of temporal regression results in comparison to the baselines. Orange color stands for temporal transfer learning ($\beta = 0.01$), the blue color corresponds to baseline 1 ($\beta = 0$), the green color stands for baseline 2 (all data) and the red color corresponds to the ARIMA model. For clarity, we show only MSE for the last 14 weekends of 2018. Almost in each time step, we see the improvement when compared to our method (orange color) with baseline 1 (blue color) and ARIMA model (red color). Wee also observe that some weekends (e.g., 10, 11) benefit from more data (baseline 2 – green color) while other weekends (e.g., 7, 8) benefit more from temporal transfer learning (orange color).

4.7 CONCLUSION AND OUTLOOK

We proposed a model that applies transfer learning in a time-dependent manner and applied it to the movie sales prediction task. In a set of experiments, we showed that our model can successfully transfer knowledge from prior time steps allowing continuous learning without forgetting. First, we demonstrated the improvement over the method that only uses the recent data for training showing that the parameter transfer is highly beneficial, especially for low-data settings. Second,

we obtained better results when compared to the model that uses all data for training at once. Therefore, we showed that more data is not necessarily helpful, especially for the applications where the language distribution may change over time. This furthermore confirms the assumption that temporal transfer learning allows focusing on those features that are relevant in a particular time step and therefore to capture temporal trends in the data. Third, by outperforming the time series baseline, we demonstrated that textual information can play an important role in financial forecasting. Thus, we showed small but consistent improvements when predicting movie sales with our method.

5

TASK DIFFERENCES IN DOMAIN ADAPTATION FOR LOW-RESOURCE SETTINGS: A CASE STUDY FOR PART-OF-SPEECH TAGGING AND SENTIMENT ANALYSIS

5.1 INTRODUCTION

The performance of models for core NLP problems heavily relies on the availability of large amounts of high-quality training data. However, this assumption rarely holds in real-world scenarios. Often, the training data is too restricted and specialized in domains such as medicine or law. To overcome this issue, most of the work adopts *supervised domain adaptation*: there is a large labeled training set available in the source domain and an amount of labeled target data that is insufficient for training a high-performing model on its own. Other methods, referred in the literature as *unsupervised domain adaptation* methods, make use of the labeled data only in the source domain. Both approaches are not directly applicable when there is no access to the labeled data in the source domain. Another challenge is the lack of training data for the domains that tend to be specific as compared to generic language use, e.g., social media or medical domains. In real-world scenarios, companies often do not have enough labeled data for their specialized tasks. They are forced to use costly professional data labeling services. The focus of this work is, therefore, to study and evaluate techniques of domain adaptation in low-resource settings for different tasks where there exists *only a small amount of labeled training data in the target domain*. In our work, we look at two specific NLP applications, sentiment analysis and Part-Of-Speech (POS) tagging, and evaluate how different domain adaptation methods fare on these tasks.

Several methods have been proposed for domain adaptation with limited training data in the target domain and no labeled data in the source domain. Many of these works focus on word embeddings (Bojanowski et al., 2016; Mikolov et al., 2013b; Pennington, Socher, and Manning, 2014) learned from large unlabeled corpora. This is one of the promising applications in transfer learning because the prior knowledge captured in embeddings can be transferred to downstream tasks with small labeled data sets. To capture domain specific characteristics, domain specific as well as domain adapted word embeddings (Sarma, 2018) have been proposed. Other ways to include information are tuning pre-trained off-the-shelf word embeddings and linear mapping (Bojanowski et al., 2019). Embedding models

like Word2Vec (Mikolov et al., 2013b), Glove (Pennington, Socher, and Manning, 2014) and FastText (Bojanowski et al., 2016) assign a single vector representation to a word independent of the context. However, the meaning of a word changes according to the context in which it is used. Recently developed contextualized embeddings like BERT (Devlin et al., 2019) or Flair (Akbik, Blythe, and Vollgraf, 2018) address this limitation by allowing to generate context-dependent representations. They furthermore allow fine-tuning all of the parameters using labeled data from the downstream tasks. For this reason, contextualized embeddings have shown tremendous success in transfer learning – they outperformed many task-specific architectures, achieving state-of-the-art performance on many sentence-level and token-level tasks.

In our work, we find that the choice of the domain adaptation method depends on the task itself; domain adaptation methods that focus on syntax and morphology are better suited for POS tagging while for sentiment analysis, semantic information is more relevant. For example, domain adapted word embeddings are only beneficial for sentiment analysis, but for POS tagging, it is better to use task-specific embeddings. Besides, further fine-tuning of word embeddings hurts the performance of sentiment analysis though it is useful for POS tagging. Surprisingly, despite the success of domain adapted word embeddings in deep models, the shallow bag-of-words approach still outperforms these methods in low-resource settings. The challenging part about domain adapted word embeddings is again the availability of domain specific data to construct them. What is not surprising is the fact that contextualized embedding models beat all other domain adaptation methods for both tasks. However, sentiment analysis benefits much more from transfer learning than POS tagging, i.e., the performance of sentiment analysis depends on the amount of data used for pre-training the contextualized model, yet it is not essential for POS tagging at all. Hence, instead of domain adapted embeddings, we recommend to use the bag-of-words model for sentiment analysis and task-specific embeddings for POS tagging. Many NLP application scenarios have resource constraints, e.g., when they have to deployed on the edge. If there are no resource constraints, then contextualized embedding methods are superior for domain adaptation in low-resource scenarios for both tasks – for sentiment analysis, it is better to use models trained on large unlabeled data, and for POS tagging, it is sufficient to apply models trained on smaller corpora like BERT base.

Domain Adaptation Setting (resource)	D_t labeled		
	none	small	large
Supervised – high resource	☐	☐	☑
Semi-supervised – low resource	☐	☑	☐
Unsupervised – zero resource	☑	☐	☐

Table 13: Different settings of domain adaptation (DA) including source domain D_s or/and target domain D_t labeled data.

5.2 OVERVIEW OF METHODS

5.2.1 *Categorization of Domain Adaptation Techniques*

In general, a domain consists of a joint distribution over a feature space and a label set (Pan et al., 2010a). Differences between domains can be characterized by their marginal distribution over the feature space and label set. So the training and test data must be from one domain to achieve good performance. The quality of the performance also depends on the amount of training data. Since specific domains often lack the availability of training data, *domain adaptation (DA)* techniques have been proposed to facilitate the transfer of knowledge from one domain (source) to a specialized domain (target). They have been successfully applied for different NLP tasks such as sentiment analysis (Glorot, Bordes, and Bengio, 2011; Pan et al., 2010b; Sarma, Liang, and Sethares, 2019), POS tagging (Astudillo et al., 2015; Schnabel and Schütze, 2014) or Named Entity Recognition (NER) (Newman-Griffis and Zirikly, 2018). Most of these methods employ *transfer learning*, one of the promising techniques for domain adaptation.

Table 13 shows our categorization of domain adaptation techniques according to *domain adaptation setting* or *resource availability* (high, low and zero resource). *Supervised domain adaptation* DA_{sup} uses the knowledge from labeled source domain D_s and applies this knowledge to a target domain D_t, given the labels in both domains (Chelba and Acero, 2006; Chen et al., 2012; Daumé III, 2009; Pan et al., 2010b). *Semi-supervised adaptation* addresses the problem that the labeled data may be too sparse to build a good classifier, by making use of a large amount of unlabeled data (in D_s and/or in D_t) and a small amount of labeled data in D_t (Pan et al., 2010a). *Unsupervised domain adaptation* refers to training a model on labeled data from a source domain D_s and applying it on a target domain, with access to unlabeled data in the target domain D_t (Blitzer, McDonald, and Pereira, 2006; Ganin et al., 2016; Glorot, Bordes, and Bengio, 2011; Han and Eisenstein, 2019; Miller, 2019; Raina et al., 2007; Yang and Eisenstein, 2015; Zhang, Li, and Ogunbona, 2019). Since we are interested in the methods for the

low-data scenario, we consider in this work only the second setting – semi-supervised domain adaptation. We also assume that we do not have labeled data in the source domain D_s because real-world scenarios may lack the training data in the source domain as well.

5.2.2 *Semi-Supervised Domain Adaptation*

To handle the case that domains and distributions used in training and testing are different, transfer learning approaches have been proposed. Most of these approaches apply embeddings, either word embeddings (Bojanowski et al., 2016; Mikolov et al., 2013b; Pennington, Socher, and Manning, 2014) or contextualized embeddings (Akbik, Blythe, and Vollgraf, 2018; Devlin et al., 2019; Liu et al., 2019). They rely on distributed vector representations that information about syntactic and semantic word relationships. Word embeddings are static word vectors: the same word will always have the same representation regardless of the context where it occurs. Contextualized embeddings make use of some language model to help to model the representation of a word and take into consideration the context of the word. The latent knowledge presented in embeddings is then transferred to downstream tasks with small labeled data sets. Since these representations capture different properties of language, they are crucial for domain adaptation in low-resource settings. An overview of the embedding models for semi-supervised domain adaptation and how they are used to transfer knowledge can be found in the Table 14.

The representations that are learned during unsupervised learning on unlabeled data can either be frozen or further optimized during supervised learning on labeled data. We refer to further optimization as fine-tuning. In the following, we give an overview of these embedding models, either "frozen" or fine-tuned.

Generic Word Embeddings. The classical solution to solve the domain adaptation problem is fine-tuning of generic word vectors, i.e., initializing the parameters with word embeddings that are pre-trained on a large corpus of source domain D_s. This method was applied as a baseline in (Bojanowski et al., 2019; Sarma, Liang, and Sethares, 2019) for text classification, in (Astudillo et al., 2015) for POS tagging and in (Newman-Griffis and Zirikly, 2018) for low-resource medical NER. The disadvantage of this approach is that aggressive fine-tuning may lead to loss of information from the original dataset (Bojanowski et al., 2019) and to overfitting when the training is performed with scarce and noisy data (Astudillo et al., 2015).

Domain Specific Word Embeddings. Domain specific word embeddings can be trained from scratch on a large amount of unlabeled data from the target domain D_t. For some domains, there already exist pre-trained models trained on specific data, e.g., Glove Twitter

Word Embeddings	Unlabeled Resources			Fine Tuning
	D_s	D_t	Lexicons/ Ontologies	
generic e.g., Glove (Pennington, Socher, and Manning, 2014), Word2Vec (Mikolov et al., 2013b)	☑	☐	☐	☑
domain specific e.g., (Shen et al., 2018b)	☐	☑	☐	☐
cross-domain e.g., (Yang, Lu, and Zheng, 2017)	☑	☑	☐	☐
refined e.g., (Faruqui et al., 2015) (Rothe and Schütze, 2017)	☑	☐	☑	☐
refined BioNLP e.g., (Patel et al., 2017) (Boag and Kané, 2017) (Ling et al., 2017)	☑	☐	☑	☐
domain adapted : KCCA (Sarma, Liang, and Sethares, 2019)	☑	☑	☐	☐
domain adapted: Linear Transformation (Bojanowski et al., 2019)	☑	☑	☐	☐
contextualized e.g., BERT (Devlin et al., 2019) RoBERTa (Liu et al., 2019) Flair (Akbik, Blythe, and Vollgraf, 2018)	☑	☐	☐	☑

Table 14: Embedding methods that use different unlabeled resources during unsupervised learning: source domain D_s data and/or target domain D_t data and/or lexicons/ontologies. They can be applied for semi-supervised domain adaptation during supervised learning on labeled data, i.e., either "frozen" or by further fine-tuning.

embeddings[1] or biomedical word embeddings BioWord2Vec (Yijia et al., 2019). Domain specific word embeddings are usually used as input features for downstream tasks where embedding weights stay fix. To achieve good performance, these embeddings should have a good quality, e.g., by training them on large corpora with good coverage of the domain.

Cross-domain Word Embeddings. In this approach, word embeddings are learned by incorporating the knowledge from both domains D_s and D_t simultaneously (He et al., 2018; Yang, Lu, and Zheng, 2017). Here, the model tries to differentiate between general and domain specific terms, by minimizing the distance between the source and the target instances in an embedded feature space. While this method was successfully applied on a sentiment classification task (He et al., 2018; Yang, Lu, and Zheng, 2017) and low-resource medi-

1 https://nlp.stanford.edu/projects/glove/

cal NER (Newman-Griffis and Zirikly, 2018), it heavily relies on the amount of training data available. Thus, this method is not applicable if the amount of unlabeled training data is small.

Refined Word Embeddings. Refined word embeddings are word embeddings augmented with additional knowledge from specific domain D_t, normally by using lexicons and ontologies. The idea is to improve general word embeddings by using valuable information that is contained in semantic lexicons such as WordNet, FrameNet, or the Paraphrase Database. For example, Faruqui et al. (2015) and Rothe and Schütze (2017) improve word representations using relational information from WordNet. In the clinical domain, the quality of embedding models was substantially improved by using domain knowledge from ontologies such as UMLS Metathesaurus, ICD, CPT and LOINIC (Boag and Kané, 2017; Ling et al., 2017).

Domain Adapted Word Embeddings. Domain adapted word embeddings are pre-trained embeddings adapted for a specialized domain. The adaptation requires domain specific embeddings of good quality. These can be pre-trained specific word embeddings like BioWord2Vec or LSA-based word vectors calculated on domain specific data D_t (Sarma, Liang, and Sethares, 2019). Domain generic and specific word embeddings are then combined using different methods:

- *Concatenation* of the two word embeddings allowing to incorporate the information from different domains (Newman-Griffis and Zirikly, 2018; Roberts, 2016).
- *Weighted mean* of the two word vectors where word weights are word frequency or inverse document frequency, which captures whether the word is common or rare across all phrases (Belousov, Dixon, and Nenadic, 2017; Sarma, Liang, and Sethares, 2019).
- *Kernel Canonical Correlation Analysis (KCCA)* of the two word embedding spaces. For example, Sarma (2018) combines pre-trained generic word embeddings and domain specific embeddings learned by applying LSA on the domain specific corpus D_t. This allows exploiting cooccurrences and context information in the domain specific data set along with the linear properties of the generic word embedding.
- *Linear Transformation* methods that use monolingual word vector alignment. Bojanowski et al. (2019) apply this approach for the scenario when the language distribution of the data drastically changes over time. Newman-Griffis and Zirikly (2018) use similar technique for low-resource medical NER – they adapt the multilingual approach proposed in (Artetxe, Labaka, and Agirre, 2016).

CONTEXTUALIZED WORD EMBEDDINGS. Recently developed transfer learning algorithms like Flair (Akbik, Blythe, and Vollgraf, 2018) and BERT (Devlin et al., 2019) propose to produce word vector representations that dynamically change with respect to the context in which the words appear. For this reason, they turned out to be an effective replacement for static word embeddings. These are fine-tuning based models, i.e., the embedding weights can be adapted to a given target task, without substantial task-specific architecture modifications. During pre-training, the model learns a general-purpose representation of inputs, and during fine-tuning (*adaptation*), the representation is transferred to a new task. Diverse context-sensitive models have been successfully applied on various downstream NLP tasks, ranging from question-answering to sentiment analysis. *Contextual string embeddings (Flair)* (Akbik, Blythe, and Vollgraf, 2018) model words as sequences of characters by learning to predict the next character on the basis of previous characters. This framework allows to better handle rare and misspelled words as well as to model subword structures such as prefixes and endings. That is why this model is beneficial for sequence labeling problems such as NER and POS tagging. *Bidirectional Encoder Representations from Transformers (BERT)* (Devlin et al., 2019) was designed to pre-train deep bidirectional representations by using a masked language model pre-training objective. In addition to the masked language model, BERT also uses a next sentence prediction task that jointly pre-trains text-pair representations. Due to this architecture, it achieves state-of-the-art performance on a large suite of sentence-level and token-level tasks, such as question-answering, paraphrasing, POS tagging and sentiment analysis. Liu et al. (2019) present a replication study of BERT pre-training, *Robustly Optimized BERT Pre-training (RoBERTa)*, with alternative design choices and training strategies that lead to better downstream task performance. They also use more data for pre-training, thereby further improving performance on downstream tasks. While these models lead to significant improvement, operating such large models remains challenging. For this reason, Sanh et al. (2019) propose DistilBERT, a smaller, faster and lighter (i.e., distilled) model of BERT. This is a general-purpose pre-trained model that can be fine-tuned with good performance on downstream tasks, keeping the flexibility of larger models.

The success of contextualized word representations suggests that despite being trained with only a language modeling task, they learn highly transferable and task-agnostic properties of language (Ethayarajh, 2019). That is why fine-tuning of these models is an important transfer learning technique that can be used for domain adaptation in low-resource settings.

5.3 CASE STUDY

In this section, we look at two specific NLP applications, sentiment analysis and POS tagging in a low-data scenario, and evaluate how different domain adaptation methods fare on these tasks. For each of these tasks, we perform experiments on two different datasets. For sentiment analysis, the datasets do not have dedicated train/dev/test splits, so we created 80/10/10 splits. In the case of a pre-trained model like BERT, we further fine-tune it using our train/dev splits.

5.3.1 *Sentiment Analysis*

For our experiments, we use a benchmark dataset of *Movie Reviews (MR)* (Pang and Lee, 2005) and the *SemEval-2017 Twitter* dataset.[2] Following the setup of Sarma, Liang, and Sethares (2019), we randomly sample 2500 positive and 2500 negative reviews for experiments with Movie Reviews. For the Twitter dataset, we randomly choose 6000 tweets with a balanced distribution of the three class labels – positive, negative and neutral.

Bag-of-words (BoW). A classical solution to solve the sentiment analysis task is to use the linear classification model on bag-of-words representations. In our experiments, we apply *logistic regression* using scikit-learn library[3] with default hyperparameters. We represent a collection of text documents using various methods:

- **Sparse Vectors.** We represent documents as sparse vectors of the size of the vocabulary, i.e., we convert a collection of text documents to a matrix of token counts.
- **Generic Word Embeddings.** Each sentence is expressed as a weighted sum of its constituent word embeddings (Glove). Weights used are raw word counts. Glove embeddings were pre-trained on Common Crawl (840B tokens). They have dimensionality 300.
- **Domain Adapted KCCA Word Embeddings.** This method uses as input features domain adapted (DA) word embeddings, formed by aligning corresponding generic (Glove) and specific word vectors with the nonlinear KCCA approach. Specific word vectors can be created using D_t by applying Latent Semantic Analysis (LSA) (Sarma, 2018). Another possible solution is to use already pre-trained domain specific embeddings. For MR, we create 300-dimensional LSA-based word embeddings. For Twitter, we use pre-trained Twitter Glove embeddings.[4] Glove Twit-

2 http://alt.qcri.org/semeval2017/task4/
3 https://scikit-learn.org/
4 https://github.com/stanfordnlp/GloVe

ter embeddings were pre-trained on a large Twitter corpus (2B tweets, 27B tokens). They have a dimensionality of 100.

- **Domain Adapted Word Embeddings: Linear Transformation.** This approach uses domain adapted vectors created by combining generic (Glove) and domain specific vectors using monolingual word vector alignment technique (Bojanowski et al., 2019). In our experiments, we use the same domain specific word embeddings as for KCCA alignment.

Neural Networks with Generic and Domain Adapted Word Embeddings. Instead of using a shallow bag-of-words approach, we utilize generic or domain adapted word embeddings as input features for Convolutional Neural Networks (CNN) (Kim, 2014) and Bi-directional LSTMs (BiLSTM) (Conneau et al., 2017). We use the Hedwig[5] library to perform our experiments. For both architectures, we set the batch size to 32, initial learning rate to 0.01 and train the model for 30 epochs. For BiLSTM, we set the number of LSTM layers to 1. Otherwise, we utilize Hedwig's default parameters. We consider two settings: (i) The word vectors are not fine-tuned during training. (ii) The word vectors are fine-tuned during training.

Contextualized Embeddings. We train a classifier that takes the output of the [CLS] token as input. We fine-tune all pre-trained parameters. Our models are BERT (Devlin et al., 2019) and its improved versions: RoBERTa (Liu et al., 2019) and DistilBERT (Sanh et al., 2019). We use the FLAIR library[6] for our experiments. We utilize mini-batch gradient descent. Batch size is 16. We train the model for 30 epochs. The initial learning rate is 2e-5. Gradients for backpropagation were estimated using the Adam optimizer (Kingma and Ba, 2015). Otherwise, we use FLAIR's default hyperparameters.

5.3.2 *Part-Of-Speech Tagging*

In our experiments, we use the *Twitter POS* dataset (Gimpel et al., 2011), which uses the universal POS tag set composed of 21 different labels. The dataset contains 1639 annotated tweets for training, 710 tweets for tuning and 1201 tweets for testing. As a second dataset, we use a POS tagged corpus built for the biomedical domain (*BioNLP POS* dataset) (Tateisi and Tsujii, 2004). It consists of 45 different labels and follows the Penn Treebank POS tagging scheme.[7] The first 1000 sentences are used for training, 500 sentences for development and 500 sentences for testing.

Window Approach. Many different methods have been employed for POS tagging with various levels of performance. One of the best

5 https://github.com/castorini/hedwig
6 https://github.com/flairNLP/flair
7 http://www.geniaproject.org/genia-corpus/pos-annotation

POS classifiers is the *window approach*, classifiers trained on windows of text (Collobert et al., 2011). In our work, we also use the window approach, which includes features such as preceding and following context words (fourgrams) and handcrafted features to deal with unknown words. Each word is thus represented as a concatenation of the handcrafted features and the following word embeddings:

- **Generic Word Embeddings (Glove).** We use the same Glove embeddings as for sentiment analysis.
- **Syntactic Word Embeddings (SENNA).** SENNA word embeddings are improved word vectors developed especially for syntax problems (Ling et al., 2015). These embeddings have 50 dimensions.
- **Domain Adapted Word Embeddings.** In our experiments, we also apply domain adapted word embeddings: KCCA (Sarma, 2018) and linear transformation based (Bojanowski et al., 2019) embeddings. They are obtained using pre-trained Glove embeddings and pre-trained domain specific embeddings. For the BioNLP POS tagging task, we use 200-dimensional biomedical word embeddings.[8] They were pre-trained with fastText on PubMed and the clinical notes from MIMIC-III. For the Twitter POS tagging task, we use Glove Twitter embeddings as for sentiment analysis.

Window Approach with Subspace. Following Astudillo et al. (2015), we improve the window approach proposed in (Collobert et al., 2011) by using the Non-Linear Sub-space Embedding (NLSE) model. NLSE is implemented as a simple feed-forward neural network model with one single hidden layer (Astudillo et al., 2015). The number of hidden states is set to 100. Window approaches are implemented using Keras.[9]

Contextualized Embeddings. In our experiments, we apply the Flair model and also BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and DistilBERT (Sanh et al., 2019) models. We use the FLAIR[10] (Akbik, Blythe, and Vollgraf, 2018) library to perform the experiments. FLAIR allows combining ("stacking") different embedding types. Since the combination of forward and backward Flair embeddings with Glove embeddings turned out to be the best for POS tagging (Akbik, Blythe, and Vollgraf, 2018), we perform this experiment instead of using only Flair embeddings. To train all the POS tagging models, we use a BiLSTM architecture on top of a contextualized model and fine-tune all parameters. We set the batch size to 32, initial learning rate to 0.1 and train the model for 150 epochs. The

8 https://github.com/ncbi-nlp/BioSentVec
9 https://keras.io/
10 https://github.com/flairNLP/flair

number of hidden states per-layer of the LSTM is set to 256. Otherwise, we use FLAIR's default hyperparameters. For the experiments with BioBERT, we utilize BioBERT[11] with default parameters.

5.4 RESULTS

5.4.1 *Sentiment Analysis*

Table 15 demonstrates the results of different domain adaptation methods for sentiment analysis. The performance measure is accuracy. The table is interesting in several ways. Comparing bag-of-words models, we can see that sparse vectors (word counts) show the highest performance on both datasets (87.4% on MR and 83.6% on Twitter). The use of generic or domain adapted word embeddings for a bag-of-words model does not improve the accuracy at all. We also see that the simple bag-of-words method is superior to the fine-tuning of generic or domain adapted word embeddings using deep models, biLSTM (Conneau et al., 2017) and CNN (Kim, 2014). Domain adapted word embeddings are only beneficial without fine-tuning – this approach slightly improves bag-of-words results for MR (87.8% vs. 87.4%) as well as for Twitter (84.8% vs. 83.6%).

We also observe that further fine-tuning of word embeddings often decreases the accuracy. For example, "frozen" linearly transformed embeddings perform better than the fine-tuned version for both datasets in both deep models. Since these embeddings are already adapted for a specialized domain, there is no need to fine-tune them unless they are of bad quality. Otherwise, the process of finetuning leads to overfitting. In contrast, KCCA embeddings for MR perform poorly without fine-tuning (58.6% in biLSTM model and 61.2% in CNN model). To obtain these embeddings, we used an LSA-based method for constructing domain specific word embeddings. A small amount of domain specific data might have a negative influence on the quality of LSA-based embeddings and thereby also on the quality of domain adapted word embeddings. This might be a reason for such poor performance of KCCA embeddings. However, linearly transformed embeddings achieve much better results than KCCA embeddings despite the use of LSA-based embeddings (e.g., 87.8% vs. 61.2% for the CNN). Hence, we can conclude that the method of Bojanowski et al. (2019) is superior when there is a small amount of specific domain data. Analyzing the results on Twitter, we see that KCCA embeddings perform much better on this dataset than on MR (e.g., 77.5% vs. 61.2% with the CNN). For constructing Twitter domain adapted embeddings, we used Twitter Glove embeddings, pre-trained on a large unlabeled Twitter data. So this illustrates that the KCCA method can provide better results when specific embeddings are of better quality.

11 https://github.com/dmis-lab/biobert

It is not surprising that contextualized models outperform other approaches. We see that RoBERTa is a clear winner for both datasets – it achieves 91.8% accuracy on MR and 89.8% accuracy on Twitter. Its superior performance can be explained by the fact that RoBERTa uses more data for pre-training than other BERT models. As indicated in (Liu et al., 2019), RoBERTa furthermore applies different training strategies and this also might lead to better performance. As anticipated, DistilBERT provides comparable results with BERT. It was developed to decrease the training time and the number of parameters, keeping the flexibility of larger models. The experimental results on both datasets support this fact.

Taken together, experimental results suggest using the simple bag-of-words model with word counts instead of more sophisticated methods with domain adapted word embeddings. Domain adapted word embeddings (i.e., KCCA) may outperform this simple method if high-quality domain specific word embeddings are available for their construction. Otherwise, it is preferable to use the method of Bojanowski et al. (2019) to create domain adapted word embeddings. In both cases, fine-tuning of these word embeddings is not necessary.

Other findings of this study indicate that contextualized models, especially RoBERTa, beat other methods on both datasets. This model was trained on a large amount of unlabeled data, and the results demonstrate that this is highly beneficial for text classification tasks. Therefore, RoBERTa can be considered as the best choice for domain specific text classification tasks in low-resource settings.

5.4.2 *Part-Of-Speech Tagging*

Table 16 compares the performance of different domain adaptation methods applied for POS tagging. The measure is accuracy. Comparing window approaches, it can be seen that the use of syntactic word embeddings (i.e., SENNA) provides the best results for the Twitter dataset (86.42%), and is also highly beneficial for BioNLP (92.85%). Slightly better results on BioNLP are achieved using "Window+Subspace (Glove)" (93.05%). The experiments also show that the use of domain adapted word embeddings does not improve the performance. Both datasets benefit from the syntactic SENNA embeddings. This is because these embeddings can better handle morphological and syntactic aspects of the language, which is more relevant for POS tagging than semantics.

For both domains, we see further improvement of the performance when fine-tuning pre-trained vectors. For example, fine-tuning of SENNA vectors allows increasing the accuracy by about 2% on both Twitter (88.87%) and on BioNLP (95.17%). We can observe similar behaviour in all models – fine-tuning of word embeddings improves the results. However, fine-tuning of word embeddings is a good approach not

	Method	MR	Twitter
static embeddings	BoW (sparse)	**87.4**	**83.6**
	BoW (generic Glove)	81.4	62.8
	BoW (KCCA)	78.0	60.1
	BoW (Linear Transf)	81.2	60.5
	biLSTM (generic Glove)	87.2	83.1
	CNN (generic Glove)	85.2	81.5
	biLSTM (KCCA)	58.6	**84.8**
	CNN (KCCA)	61.2	77.5
	biLSTM (Linear Transf)	85.8	83.1
	CNN (Linear Transf)	**87.8**	83.3
finetuning	biLSTM (generic Glove)	**86.6**	82.6
	CNN (generic Glove)	80.2	80.0
	biLSTM (KCCA)	69.6	**83.6**
	CNN (KCCA)	73.4	83.5
	biLSTM (Linear Transf)	80.2	77.6
	CNN (Linear Transf)	82.2	82.1
	BERT	89.4	86.1
	RoBERTa	**91.8**	**89.8**
	DistilBERT	89.2	86.6

Table 15: Sentiment analysis accuracy on Movie Reviews (MR) and Twitter. Best accuracy for each dataset is bolded.

for all tasks – POS tagging benefits from it while sentiment analysis usually experiences performance drops due to overfitting.

We also see that contextualized embedding models perform much better than other methods. For example, the concatenation of Flair and Glove embeddings beats all other window approaches for both Twitter (93.1%) and BioNLP (97.7%). A reasonable explanation is that this Flair contextualizes based on surrounding context and, in addition, models words as sequences of characters. Since BioBERT was trained on a large amount of data from the biomedical domain, it achieves the best performance for the BioNLP POS tagging task (98.3%). However, BERT (98.1%), as well as RoBERTa (98%), achieve almost the same good results. Best methods for the Twitter dataset are also BERT (94.8%) and RoBERTa (95.1%). A compressed version of BERT, DistilBERT, demonstrates good results as well – it achieves accuracy of 97.8% for the BioNLP task and accuracy of 94.2% for the Twitter task. Its performance is even superior to the Flair model for both datasets.

The evidence from these experiments suggests using contextualized models for POS tagging in low-resource settings. Despite the fact

	Method	Twitter	BioNLP
static embeddings	W (Glove)	84.5	92.2
	W+Subsp (Glove)	84.5	**93.1**
	W(SENNA)	**86.4**	92.8
	W(Linear Transf)	80.6	89.7
	W (KCCA)	84.0	92.7
finetuning	W (Glove)	87.5	94.9
	W+Subsp (Glove)	87.5	**95.2**
	W (SENNA)	**88.8**	94.3
	W(Linear Transf)	88.5	93.9
	W(KCCA)	87.6	94.0
	BioBERT	–	**98.3**
	BERT	94.8	98.1
	Flair (+Glove)	93.1	97.7
	RoBERTa	**95.1**	98.0
	DistilBERT	94.2	97.8

Table 16: POS tagging accuracy on Twitter and BioNLP. W = Window Approach

that the contextualized string embedding model Flair can better handle unknown and rare words as well as model subword structures such as prefixes and endings, *contextualized word embedding* models like BERT and RoBERTa outperform it. Other findings of this study indicate that the *task specific* word embeddings SENNA surpass both generic and domain adapted embeddings and that fine-tuning of word embeddings further improves the results.

5.5 DISCUSSION AND CONCLUSION

The observations from our experiments indicate that different tasks should be treated differently, e.g., sentiment analysis benefits from domain adapted word embeddings while for Part-Of-Speech (POS), it is better to use unadapted task-specific word embeddings. This is because these task-specific embeddings can better handle morphological and syntactic aspects, and for POS tagging this is more important than semantics. Surprisingly, simple methods (like bag-of-words) outperform BiLSTM and CNN models with domain adapted word embeddings. Domain adapted word embeddings may beat this simple model but only if domain specific word embeddings used for their construction are of high quality. Besides, further fine-tuning of word embeddings often decreases the accuracy for sentiment analysis

while it improves the results for POS tagging in all cases. Contextualized embedding methods are superior to all other domain adaptation methods across tasks – for sentiment analysis, it is preferable to use large models (e.g, RoBERTa) and for POS tagging, smaller models are sufficient (e.g. BERT base).

6 CONCLUSION

In this thesis, we summarized the mechanisms and the strategies of domain adaptation in Natural Language Processing describing a large number of approaches and related works. Domain adaptation is classified into three different settings: supervised, semi-supervised, and unsupervised. The oldest approaches focus on the supervised methods while recently proposed approaches apply semi-supervised and unsupervised methods. The latter may attract more and more attention in the future due to the lack of training data in specialized domains.

To investigate the problem of domain adaptation, we performed several experiments from different perspectives. At first, we conducted domain adaptation in finance: we automatically adapted sentiment dictionaries for predicting financial outcomes. We demonstrated that the automatically adapted sentiment dictionary outperforms the previous state of the art in predicting excess return and volatility. This shows that automatic adaptation performs better than manual adaptation. This is due to the fact that some words can be misclassified by humans because the prevalent context is not always obvious to the annotator. Our quantitative and qualitative study confirmed this observation and provided insight into the semantics of our dictionaries. This has led us to conclude that the annotation based on an expert's a priori belief about a word's meaning can be incorrect – annotation should be performed based on the word's contexts in the target domain.

Next, we investigated whether the language change over time can significantly impact the prediction. For example, the language of social media is changing very rapidly – neologisms, named entities, and trends are constantly emerging. To account for these changes, we developed the temporal transfer learning method to model the temporal dynamics in the document collection. In a set of experiments, we showed that this technique significantly improved the prediction of movie sales from discussions on Reddit forums. First, we demonstrated that our method improves performance when compared to the method that only uses recent data for training. This illustrates the success and importance of parameter transfer from previous models. Second, we showed the benefit of temporal transfer learning over the autoregressive model. This illustrates the value of textual information for financial forecasting and the advantage of our method if the data has a limited-time series history. Third, we showed that our approach outperforms a model that uses the same training data, but combines

it "naively" by using it all at once without taking time-dependence into account. This shows that more data is not necessarily helpful, especially for the applications where the language distribution may change over time. This furthermore verifies the assumption that temporal transfer learning can capture temporal trends in the data by focusing on those features that are relevant in a particular time step, i.e., we obtained more robust models preventing overfitting. All in all the proposed approach requires little retraining and only storing the previous model while discarding the old data. Thus, our model can be eventually deployed in a live system for real-time forecasting.

At last, we conducted experiments to evaluate the performance of domain adaptation methods in a low-resource setting. In particular, we applied different approaches for Sentiment Analysis and Part-of-Speech Tagging problems for Twitter, Movie Reviews, and the Clinical Domain. We found that contextualized models beat other word embedding based methods across both tasks. However, the choice of contextualized model depends on the task itself. Text classification tasks benefit more from models that are pre-trained on a large amount of unlabeled data while for POS tagging it is sufficient to apply smaller models like BERT base. Other observations indicate that simple methods like bag-of-words surpass more sophisticated methods with domain-adapted word embeddings. Domain-specific word embeddings are also outperformed in the POS tagging task, i.e., by using task-specific embeddings. Thus, if the amount of training data in a specialized domain is low, one can either use shallow methods like bag-of-words or task-specific embeddings or carefully choose a contextualized embedding method.

We believe two future directions are promising in the near term:

(i) *Online NLP techniques.* Little work has been done for exploring online NLP methods so far. Online algorithms can be advantageous in many situations for NLP. For example, they can be useful in finance for monitoring the market using textual information like news. Another application is adapting models incrementally as to new data becomes available, e.g., for such tasks as spam filtering, customer profiling for marketing, or categorization of textual information. This will make it possible to reduce the amount of retraining that is necessary – only the previous parameters have to be stored, not the data. Furthermore, as found in our experiments, online methods can account for language change over time – new words, named entities, and trends.

(ii) *Domain adapted contextualized models.* Contextualized word embeddings provide a strong performance across a wide range of NLP tasks by pre-training on large corpora of unlabeled text. However, the texts used to build pre-trained contextualized word embedding models are from a general domain. Thus, an interesting topic for future research is the adaptation of contextualized models for a specific domain, especially in low-data scenarios.

To sum up the foregoing, one must agree that domain adaptation is an important research topic in computational linguistics since every Natural Language Processing system is domain-specific and requires to account for the domain information.

APPENDIX

A.1 EXCESS RETURN REGRESSION RESULTS FOR MULTIPLE TEXT VARIABLES

var	coeff	std coeff	t	R^2
$H4N_{RE}$	-0.88**	*-0.264*	**-2.19**	1.05
neg_{lm}	0.062	0.024	0.48	
$H4N_{RE}$	-0.739**	*-0.221*	**-2.23**	1.05
all_{lm}	-0.008	-0.008	-0.21	
$H4N_{RE}$	-0.836**	*-0.25*	**-2.15**	1.05
neg_unc_{lm}	0.027	0.016	0.28	
$H4N_{RE}$	-0.755**	*-0.226*	**-2.56**	1.05
neg_lit_{lm}	-0.003	-0.004	-0.12	

*p $\leqslant$ 0.05, **p $\leqslant$ 0.01, ***p $\leqslant$ 0.001

Table 17: This table shows results for regressions that combine $H4N_{RE}$ with single-feature manual L&M lists.

var	coeff	std coeff	t	R^2
neg_{lm}	-0.202**	-0.080	**-2.56**	1.02
neg_{RE}	-0.37***	-0.111	**-2.96**	1.02
neg_{ADD}	-0.033	-0.0231	-1.03	1.00
neg_{lm}	-0.0607	-0.0242	-0.38	1.02
neg_{RE}	-0.274	-0.0822	-1.11	
neg_{RE}	-0.416***	*-0.124*	**-2.85**	1.02
neg_{ADD}	0.0298	0.0208	0.80	
neg_{lm}	-0.0421	-0.0168	-0.27	1.02
neg_{RE}	-0.346	-0.1037	-1.35	
neg_{ADD}	0.0277	0.0193	0.76	

*p $\leqslant$ 0.05, **p $\leqslant$ 0.01, ***p $\leqslant$ 0.001

Table 18: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the negative category.

var	coeff	std coeff	t	R^2
unc_{lm}	-0.215*	-0.064	**-1.91**	1.01
unc_{RE}	-0.377***	-0.075	**-2.77**	1.02
unc_{ADD}	0.0217	0.0065	0.21	1.00
unc_{lm}	0.209	0.0626	0.45	1.01
unc_{RE}	-0.668	*-0.133*	-1.05	
unc_{RE}	-0.643***	-0.128	**-3.14**	1.03
unc_{ADD}	0.198	0.0594	1.42	
unc_{lm}	-0.233	-0.0699	-0.42	1.03
unc_{RE}	-0.368	-0.0736	-0.54	
unc_{ADD}	0.234	0.0702	1.42	

*p $\leqslant$ 0.05, **p $\leqslant$ 0.01, ***p $\leqslant$ 0.001

Table 19: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the uncertain category.

A.2 VOLATILITY REGRESSION RESULTS FOR MULTIPLE TEXT VARIABLES

var	coeff	std coeff	t	R^2
lit_{lm}	-0.0291	-0.026	-0.83	1.00
lit_{RE}	-0.056	-0.028	-0.55	1.02
lit_{ADD}	-0.0195	-0.0156	-0.70	1.00
lit_{lm}	-0.0759	*-0.0683*	-0.95	1.00
lit_{RE}	0.154	0.077	0.67	
lit_{RE}	-0.0261	-0.0130	-0.20	1.00
lit_{ADD}	-0.0136	-0.0108	-0.39	
lit_{lm}	-0.0753	-0.0677	-0.94	1.00
lit_{RE}	0.155	0.0775	0.66	
lit_{ADD}	-0.00107	-0.0008	-0.03	

*p $\leqslant$ 0.05, **p $\leqslant$ 0.01, ***p $\leqslant$ 0.001

Table 20: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the litigious category.

var	coeff	std coeff	t	R^2
$H4N_{RE}$	0.748***	*0.224*	**4.44**	60.3
neg_{lm}	-0.096*	-0.038	**-2.55**	
$H4N_{RE}$	0.741***	*0.222*	**4.30**	60.3
all_{lm}	-0.0438**	-0.0481	**-2.95**	
$H4N_{RE}$	0.696***	*0.208*	**4.88**	60.3
neg_unc_{lm}	-0.054	-0.032	-1.86	
$H4N_{RE}$	0.693***	*0.207*	**4.24**	60.3
neg_lit_{lm}	-0.034**	-0.037	**-2.70**	

*p $\leqslant$ 0.05, **p $\leqslant$ 0.01, ***p $\leqslant$ 0.001

Table 21: This table shows results for regressions that combine $H4N_{RE}$ with single-feature manual L&M lists.

var	coeff	std coeff	t	R^2
neg_{lm}	0.118***	0.0472	**3.30**	60.1
neg_{RE}	0.219***	0.0657	**3.57**	60.1
neg_{ADD}	0.032***	0.0224	**4.06**	60.0
neg_{lm}	0.0014	0.0005	0.02	60.1
neg_{RE}	0.217*	0.065	**1.96**	
neg_{RE}	0.233**	*0.0699*	**2.96**	60.1
neg_{ADD}	-0.0087	-0.006	-0.65	
neg_{lm}	0.00069	0.0002	0.01	60.1
neg_{RE}	0.232*	0.0696	**1.97**	
neg_{ADD}	-0.0087	-0.006	-0.66	

*p $\leqslant$ 0.05, **p $\leqslant$ 0.01, ***p $\leqslant$ 0.001

Table 22: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the negative category.

var	coeff	std coeff	t	R^2
unc_{lm}	0.119*	0.0356	**2.25**	60.0
unc_{RE}	0.167*	0.0334	**2.30**	60.0
unc_{ADD}	-0.013	-0.0039	-0.17	60.0
unc_{lm}	0.0432	0.012	0.28	60.0
unc_{RE}	0.112	0.0224	0.53	
unc_{RE}	0.222***	0.0444	**3.48**	60.1
unc_{ADD}	-0.088	-0.0263	-1.09	
unc_{lm}	0.151	*0.0453*	1.11	60.1
unc_{RE}	0.0419	0.0083	0.20	
unc_{ADD}	-0.111	-0.0332	-1.41	

*p $\leqslant$ 0.05, **p $\leqslant$ 0.01, ***p $\leqslant$ 0.001

Table 23: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the uncertain category.

var	coeff	std coeff	t	R^2
lit_{lm}	-0.0081	-0.0073	-0.62	60.0
lit_{RE}	0.0080	0.004	0.20	60.0
lit_{ADD}	0.028	0.0224	1.07	60.0
lit_{lm}	-0.0635**	-0.057	**-2.93**	60.0
lit_{RE}	0.181*	*0.0905*	**2.46**	
lit_{RE}	-0.362	-0.181	-0.91	60.0
lit_{ADD}	0.041	0.0328	1.50	
lit_{lm}	-0.087***	-0.078	**-3.65**	60.1
lit_{RE}	0.174*	0.087	**2.42**	
lit_{ADD}	0.066*	0.0528	**2.23**	

*p $\leqslant$ 0.05, **p $\leqslant$ 0.01, ***p $\leqslant$ 0.001

Table 24: This table shows results for regressions that combine RE, ADD and L&M dictionaries for the litigious category.

BIBLIOGRAPHY

Akbik, Alan, Duncan Blythe, and Roland Vollgraf (Aug. 2018). "Contextual String Embeddings for Sequence Labeling." In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1638–1649. URL: https://www.aclweb.org/anthology/C18-1139 (cit. on pp. 14, 58, 60, 61, 63, 66).

Amir, Silvio, Wang Ling, Ramón Fernández Astudillo, Bruno Martins, Mário J. Silva, and Isabel Trancoso (2015). "INESC-ID: A Regression Model for Large Scale Twitter Sentiment Lexicon Induction." In: *SemEval@NAACL-HLT*. The Association for Computer Linguistics, pp. 613–618 (cit. on p. 28).

Antweiler, Werner and Murray Frank (Feb. 2004). "Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards." In: *Journal of Finance* 59, pp. 1259–1294. DOI: 10.2139/ssrn.282320 (cit. on p. 25).

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre (Nov. 2016). "Learning principled bilingual mappings of word embeddings while preserving monolingual invariance." In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 2289–2294. DOI: 10.18653/v1/D16-1250. URL: https://www.aclweb.org/anthology/D16-1250 (cit. on p. 62).

Astudillo, Ramon, Silvio Amir, Wang Ling, Mário Silva, and Isabel Trancoso (July 2015). "Learning Word Representations from Scarce and Noisy Data with Embedding Subspaces." 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)*. Beijing, China: Association for Computational Linguistics, pp. 1074–1084. DOI: 10.3115/v1/P15-1104. URL: https://www.aclweb.org/anthology/P15-1104 (cit. on pp. 59, 60, 66).

Axelrod, Amittai, Xiaodong He, and Jianfeng Gao (July 2011). "Domain Adaptation via Pseudo In-Domain Data Selection." In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, pp. 355–362. URL: https://www.aclweb.org/anthology/D11-1033 (cit. on p. 9).

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani (2010). "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." In: *LREC*. Ed. by Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Ste-

lios Piperidis, Mike Rosner, and Daniel Tapias. European Language Resources Association. ISBN: 2-9517408-6-7. URL: http://nmis.isti.cnr.it/sebastiani/Publications/LREC10.pdf (cit. on p. 28).

Belo, Frederico, Jun Li, Xiaoji Lin, and Xiaofei Zhao (2016). "Complexity and Information Content of Financial Disclosures: Evidence from Evolution of Uncertainty Following 10-K Filings." In: *SSRN* (cit. on p. 38).

Belousov, Maksim, William Dixon, and Goran Nenadic (Jan. 2017). "Using an Ensemble of Generalised Linear and Deep Learning Models in the SMM4H 2017 Medical Concept Normalisation Task." In: (cit. on p. 62).

Bengio, Yoshua (Jan. 2009). "Learning Deep Architectures for AI." In: *Found. Trends Mach. Learn.* 2.1, 1–127. ISSN: 1935-8237. DOI: 10.1561/2200000006. URL: https://doi.org/10.1561/2200000006 (cit. on p. 21).

Blitzer, John, Ryan McDonald, and Fernando Pereira (2006). "Domain adaptation with structural correspondence learning." In: *Proceedings of the 2006 conference on empirical methods in natural language processing*. Association for Computational Linguistics, pp. 120–128 (cit. on pp. 18, 19, 28, 59).

Boag, Willie and Hassan Kané (2017). "AWE-CM Vectors: Augmenting Word Embeddings with a Clinical Metathesaurus." In: *CoRR* abs/1712.01460. arXiv: 1712.01460. URL: http://arxiv.org/abs/1712.01460 (cit. on pp. 61, 62).

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2016). "Enriching Word Vectors with Subword Information." In: *arXiv preprint arXiv:1607.04606* (cit. on pp. 57, 58, 60).

Bojanowski, Piotr, Onur Celebi, Tomas Mikolov, Edouard Grave, and Armand Joulin (2019). "Updating Pre-trained Word Vectors and Text Classifiers using Monolingual Alignment." In: *CoRR* abs/1910.06241. arXiv: 1910.06241. URL: http://arxiv.org/abs/1910.06241 (cit. on pp. 12, 57, 60–62, 65–68).

Bollegala, Danushka, Takanori Maehara, and Ken-ichi Kawarabayashi (July 2015). "Unsupervised Cross-Domain Word Representation Learning." 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)*. Beijing, China: Association for Computational Linguistics, pp. 730–740. DOI: 10.3115/v1/P15-1071. URL: https://www.aclweb.org/anthology/P15-1071 (cit. on p. 12).

Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer (2002). "SMOTE: Synthetic Minority Over-sampling Technique." In: *Journal of Artificial Intelligence Research* 16, pp. 321–357 (cit. on p. 9).

Chelba, Ciprian and Alex Acero (2006). "Adaptation of maximum entropy capitalizer: Little data can help a lot." In: *Computer Speech & Language* 20.4, pp. 382–399 (cit. on pp. 10, 28, 59).

Chen, Minmin, Zhixiang Xu, Kilian Weinberger, and Fei Sha (2012). "Marginalized denoising autoencoders for domain adaptation." In: *arXiv preprint arXiv:1206.4683* (cit. on pp. 22, 23, 28, 59).

Chen, Minmin, Kilian Q. Weinberger, Zhixiang Xu, and Fei Sha (Jan. 2015). "Marginalizing Stacked Linear Denoising Autoencoders." In: *J. Mach. Learn. Res.* 16.1, 3849–3875. ISSN: 1532-4435 (cit. on pp. 21, 22).

Chen, Xilun, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Q. Weinberger (2018). "Adversarial Deep Averaging Networks for Cross-Lingual Sentiment Classification." In: *Trans. Assoc. Comput. Linguistics* 6, pp. 557–570. URL: https://transacl.org/ojs/index.php/tacl/article/view/1413 (cit. on p. 23).

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa (Nov. 2011). "Natural Language Processing (Almost) from Scratch." In: *J. Mach. Learn. Res.* 999888, pp. 2493–2537. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=2078183.2078186 (cit. on p. 66).

Conneau, Alexis, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes (Sept. 2017). "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 670–680. DOI: 10.18653/v1/D17-1070. URL: https://www.aclweb.org/anthology/D17-1070 (cit. on pp. 12, 65, 67).

Dai, Zihang, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov (2019). "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context." In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Ed. by Anna Korhonen, David R. Traum, and Lluís Màrquez. Association for Computational Linguistics, pp. 2978–2988. DOI: 10.18653/v1/p19-1285. URL: https://doi.org/10.18653/v1/p19-1285 (cit. on p. 16).

Daumé III, Hal (2009). "Frustratingly easy domain adaptation." In: *arXiv preprint arXiv:0907.1815* (cit. on pp. 7, 10, 11, 28, 59).

Daumé III, Hal and Daniel Marcu (2006). "Domain Adaptation for Statistical Classifiers." In: *J. Artif. Intell. Res.* 26, pp. 101–126. URL: http://dblp.uni-trier.de/db/journals/jair/jair26.html#DaumeM06 (cit. on p. 10).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Confer-*

ence of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/n19-1423. URL: https://doi.org/10.18653/v1/n19-1423 (cit. on pp. 15, 16, 58, 60, 61, 63, 65, 66).

Dredze, Mark and Koby Crammer (Oct. 2008). "Online Methods for Multi-Domain Learning and Adaptation." In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 689–697. URL: https://www.aclweb.org/anthology/D08-1072 (cit. on p. 44).

Ethayarajh, Kawin (Nov. 2019). "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 55–65. DOI: 10.18653/v1/D19-1006. URL: https://www.aclweb.org/anthology/D19-1006 (cit. on p. 63).

Fama, Eugene F. and Kenneth R. French (1993). "Common risk factors in the returns on stocks and bonds." In: *Journal of Financial Economics* 33.1, pp. 3 –56. ISSN: 0304-405X. DOI: https://doi.org/10.1016/0304-405X(93)90023-5. URL: http://www.sciencedirect.com/science/article/pii/0304405X93900235 (cit. on p. 30).

Fama, Eugene F and James D MacBeth (1973). "Risk, return, and equilibrium: Empirical tests." In: *Journal of political economy* 81.3, pp. 607–636 (cit. on p. 29).

Faruqui, Manaal, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith (2015). "Retrofitting Word Vectors to Semantic Lexicons." In: *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*. Ed. by Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar. The Association for Computational Linguistics, pp. 1606–1615. DOI: 10.3115/v1/n15-1184. URL: https://doi.org/10.3115/v1/n15-1184 (cit. on pp. 61, 62).

Fellbaum, Christiane, ed. (1998). *WordNet: an electronic lexical database*. MIT Press (cit. on p. 28).

Finkel, Jenny Rose and Christopher D. Manning (June 2009). "Hierarchical Bayesian Domain Adaptation." In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, Colorado: Association for Computational Linguistics, pp. 602–

610. URL: https://www.aclweb.org/anthology/N09-1068 (cit. on p. 10).

Gama, João, Indrundefined Žliobaitundefined, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia (Mar. 2014). "A Survey on Concept Drift Adaptation." In: *ACM Comput. Surv.* 46.4. ISSN: 0360-0300. DOI: 10.1145/2523813. URL: https://doi.org/10.1145/2523813 (cit. on p. 44).

Ganin, Yaroslav, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky (2016). "Domain-adversarial training of neural networks." In: *The Journal of Machine Learning Research* 17.1, pp. 2096–2030 (cit. on pp. 22, 28, 59).

Gelbach, Jonah B, Doug Miller, et al. (2009). *Robust Inference with Multi-way Clustering*. Tech. rep. National Bureau of Economic Research (cit. on p. 29).

Gimpel, Kevin, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith (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*. HLT '11. Portland, Oregon: Association for Computational Linguistics, pp. 42–47. ISBN: 978-1-932432-88-6. URL: http://dl.acm.org/citation.cfm?id=2002736.2002747 (cit. on p. 65).

Gliozzo, Alfio Massimiliano and Carlo Strapparava (2009). *Semantic Domains in Computational Linguistics*. Springer, pp. I–IX, 1–131. ISBN: 978-3-540-68156-4 (cit. on p. 1).

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). "Domain adaptation for large-scale sentiment classification: A deep learning approach." In: *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 513–520 (cit. on pp. 21, 28, 59).

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press (cit. on pp. 19–22).

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative Adversarial Nets." In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf (cit. on p. 22).

Gruhl, Daniel, R. Guha, Ravi Kumar, Jasmine Novak, and Andrew Tomkins (2005). "The Predictive Power of Online Chatter." In: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. KDD '05. Chicago, Illinois, USA: ACM, pp. 78–87. ISBN: 1-59593-135-X. DOI: 10.1145/

1081870.1081883. URL: http://doi.acm.org/10.1145/1081870.1081883 (cit. on pp. 43, 45, 47).

Hamilton, James (1994). *D.(1994), Time Series Analysis* (cit. on p. 51).

Hamilton, William L., Kevin Clark, Jure Leskovec, and Dan Jurafsky (2016). "Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora." In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. Ed. by Jian Su, Xavier Carreras, and Kevin Duh. The Association for Computational Linguistics, pp. 595–605. DOI: 10.18653/v1/d16-1057. URL: https://doi.org/10.18653/v1/d16-1057 (cit. on pp. 28, 36, 37).

Han, Xiaochuang and Jacob Eisenstein (2019). "Unsupervised Domain Adaptation of Contextualized Embeddings: A Case Study in Early Modern English." In: *CoRR* abs/1904.02817. arXiv: 1904.02817. URL: http://arxiv.org/abs/1904.02817 (cit. on pp. 17, 59).

Hanks, Patrick (2000). "Do word meanings exist?" In: *Computers and the Humanities* 34.1–2, pp. 205–215. URL: http://www.coli.uni-sb.de/~kowalski/senseval/hanks.pdf (cit. on p. 1).

Hardoon, David R., Sándor Szedmák, and John Shawe-Taylor (2004). "Canonical Correlation Analysis: An Overview with Application to Learning Methods." In: *Neural Computation* 16.12, pp. 2639–2664. DOI: 10.1162/0899766042321814. URL: https://doi.org/10.1162/0899766042321814 (cit. on p. 12).

Harris, Zellig (1954). "Distributional structure." In: *Word* 10.23, pp. 146–162 (cit. on p. 1).

Hatzivassiloglou, Vasileios and Kathleen R. McKeown (1997). "Predicting the Semantic Orientation of Adjectives." In: *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*. EACL '97. Madrid, Spain: Association for Computational Linguistics, pp. 174–181. DOI: 10.3115/979617.979640. URL: https://doi.org/10.3115/979617.979640 (cit. on p. 27).

He, Ruidan, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier (2018). "Adaptive Semi-supervised Learning for Cross-domain Sentiment Classification." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 3467–3476. DOI: 10.18653/v1/D18-1383. URL: https://www.aclweb.org/anthology/D18-1383 (cit. on p. 61).

Herman Vimala, Z.T.B.N., B. Nerlich, D.D. Clarke, Z. Todd, and V. Herman (2003). *Polysemy: Flexible Patterns of Meaning in Mind and Language*. Trends in linguistics / Studies and monographs: Studies and monographs. Mouton de Gruyter. ISBN: 9783110176162. URL: https://books.google.de/books?id=Hjq_mOzljOIC (cit. on p. 1).

Ho, Joyce C. and Yubin Park (2017). "Learning from different perspectives: Robust cardiac arrest prediction via temporal transfer learning." In: *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jeju Island, South Korea, July 11-15, 2017*, pp. 1672–1675. DOI: 10.1109/EMBC.2017.8037162. URL: https://doi.org/10.1109/EMBC.2017.8037162 (cit. on p. 44).

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory." In: *Neural Computation* 9.8, pp. 1735–1780 (cit. on p. 14).

Hoffman, Matthew, Francis R Bach, and David M Blei (2010). "Online learning for latent dirichlet allocation." In: *advances in neural information processing systems*, pp. 856–864 (cit. on p. 44).

Hotelling, Harold (Dec. 1936). "RELATIONS BETWEEN TWO SETS OF VARIATES." In: *Biometrika* 28.3-4, pp. 321–377. ISSN: 0006-3444. DOI: 10.1093/biomet/28.3-4.321. URL: https://doi.org/10.1093/biomet/28.3-4.321 (cit. on p. 12).

Howard, Jeremy and Sebastian Ruder (July 2018). "Universal Language Model Fine-tuning for Text Classification." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 328–339. DOI: 10.18653/v1/P18-1031. URL: https://www.aclweb.org/anthology/P18-1031 (cit. on pp. 13, 14).

Hu, Minqing and Bing Liu (2004). "Mining and summarizing customer reviews." In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*. Ed. by Won Kim, Ron Kohavi, Johannes Gehrke, and William DuMouchel. ACM, pp. 168–177. DOI: 10.1145/1014052.1014073. URL: https://doi.org/10.1145/1014052.1014073 (cit. on p. 46).

Huang, Sheng, Zhendong Niu, and Chongyang Shi (2014). "Automatic construction of domain-specific sentiment lexicon based on constrained label propagation." In: *Knowl.-Based Syst.* 56, pp. 191–200 (cit. on p. 27).

Huang, Zhiheng, Wei Xu, and Kai Yu (2015). "Bidirectional LSTM-CRF Models for Sequence Tagging." In: *CoRR* abs/1508.01991. arXiv: 1508.01991. URL: http://arxiv.org/abs/1508.01991 (cit. on p. 15).

Igo, Sean P. and Ellen Riloff (2009). "Corpus-based Semantic Lexicon Induction with Web-based Corroboration." In: *Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*. UMSLLS '09. Boulder, Colorado: Association for Computational Linguistics, pp. 18–26. ISBN: 978-1-932432-34-3. URL: http://dl.acm.org/citation.cfm?id=1641968.1641971 (cit. on p. 27).

Jacobusse, Gert and Cor J. Veenman (2016). "On Selection Bias with Imbalanced Classes." In: *Discovery Science - 19th International Conference, DS 2016, Bari, Italy, October 19-21, 2016, Proceedings*. Ed. by Toon Calders, Michelangelo Ceci, and Donato Malerba. Vol. 9956. Lecture Notes in Computer Science, pp. 325–340. DOI: 10.1007/978-3-319-46307-0_21. URL: https://doi.org/10.1007/978-3-319-46307-0_21 (cit. on p. 9).

Jiang, Jing and ChengXiang Zhai (June 2007). "Instance Weighting for Domain Adaptation in NLP." In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 264–271. URL: https://www.aclweb.org/anthology/P07-1034 (cit. on p. 8).

Kazemian, Siavash, Shunan Zhao, and Gerald Penn (Aug. 2016). "Evaluating Sentiment Analysis in the Context of Securities Trading." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 2094–2103. DOI: 10.18653/v1/P16-1197. URL: https://www.aclweb.org/anthology/P16-1197 (cit. on p. 45).

Kim, Yoon (Oct. 2014). "Convolutional Neural Networks for Sentence Classification." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1746–1751. DOI: 10.3115/v1/D14-1181. URL: https://www.aclweb.org/anthology/D14-1181 (cit. on pp. 65, 67).

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization." In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6980 (cit. on pp. 49, 65).

Kogan, Shimon, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith (2009). "Predicting Risk from Financial Reports with Regression." In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. NAACL '09. Boulder, Colorado: Association for Computational Linguistics, pp. 272–280. ISBN: 978-1-932432-41-1. URL: http://dl.acm.org/citation.cfm?id=1620754.1620794 (cit. on pp. 43, 45).

Kouw, Wouter M. and Marco Loog (2019). "A review of single-source unsupervised domain adaptation." In: *CoRR* abs/1901.05335. arXiv: 1901.05335. URL: http://arxiv.org/abs/1901.05335 (cit. on pp. 8–10, 18).

Le, Lei, Andrew Patterson, and Martha White (2018). "Supervised autoencoders: Improving generalization performance with unsu-

pervised regularizers." In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc., pp. 107–117. URL: http://papers.nips.cc/paper/7296-supervised-autoencoders-improving-generalization-performance-with-unsupervised-regularizers.pdf (cit. on p. 19).

Lee, Heeyoung, Mihai Surdeanu, Bill MacCartney, and Dan Jurafsky (Jan. 2014). "On the importance of text analysis for stock price prediction." English (US). In: *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*. European Language Resources Association (ELRA), pp. 1170–1175 (cit. on p. 45).

Ling, Wang, Chris Dyer, Alan W. Black, and Isabel Trancoso (2015). "Two/Too Simple Adaptations of Word2Vec for Syntax Problems." In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 1299–1304. DOI: 10.3115/v1/N15-1142. URL: https://www.aclweb.org/anthology/N15-1142 (cit. on p. 66).

Ling, Yuan, Yuan An, Mengwen Liu, Sadid A. Hasan, Ye-tian Fan, and Xiaohua Hu (2017). "Integrating extra knowledge into word embedding models for biomedical NLP tasks." In: *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*. IEEE, pp. 968–975. DOI: 10.1109/IJCNN.2017.7965957. URL: https://doi.org/10.1109/IJCNN.2017.7965957 (cit. on pp. 61, 62).

Liu, Chenghao, Steven C. H. Hoi, Peilin Zhao, and Jianling Sun (2016). "Online ARIMA Algorithms for Time Series Prediction." In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. Phoenix, Arizona: AAAI Press, pp. 1867–1873. URL: http://dl.acm.org/citation.cfm?id=3016100.3016160 (cit. on pp. 43, 45, 51).

Liu, Yang, Jimmy Huang, Aijun An, and Xiaohui Yu (2007). "ARSA: A Sentiment-Aware Model for Predicting Sales Performance Using Blogs." In: *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)* (cit. on pp. 43, 45, 47, 48).

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach." In: *CoRR* abs/1907.11692. arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692 (cit. on pp. 15, 16, 60, 61, 63, 65, 66, 68).

Loughran, Tim and Bill Mcdonald (2011). "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks." In: *The Journal of Finance* 66.1, pp. 35–65. ISSN: 1540-6261. DOI: 10.1111/j.

1540-6261.2010.01625.x. URL: http://dx.doi.org/10.1111/j.1540-6261.2010.01625.x (cit. on pp. 25, 26, 29, 30, 33, 38).

Ma, Xiaofei, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang (Nov. 2019). "Domain Adaptation with BERT-based Domain Classification and Data Selection." In: *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. Hong Kong, China: Association for Computational Linguistics, pp. 76–83. DOI: 10.18653/v1/D19-6109. URL: https://www.aclweb.org/anthology/D19-6109 (cit. on p. 17).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013a). "Distributed Representations of Words and Phrases and their Compositionality." In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Curran Associates, Inc., pp. 3111–3119. URL: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf (cit. on p. 31).

Mikolov, Tomas, Kai Chen, G.s Corrado, and Jeffrey Dean (Jan. 2013b). "Efficient Estimation of Word Representations in Vector Space." In: *Proceedings of Workshop at ICLR* 2013 (cit. on pp. 11, 13, 57, 58, 60, 61).

Miller, Timothy (June 2019). "Simplified Neural Unsupervised Domain Adaptation." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 414–419. DOI: 10.18653/v1/N19-1039. URL: https://www.aclweb.org/anthology/N19-1039 (cit. on p. 59).

Mohammad, Saif, Svetlana Kiritchenko, and Xiaodan Zhu (2013). "NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets." In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, pp. 321–327. URL: http://aclweb.org/anthology/S13-2053 (cit. on p. 27).

Moniz, Andy and Franciska de Jong (2014). "Classifying the Influence of Negative Affect Expressed by the Financial Media on Investor Behavior." In: *Proceedings of the 5th Information Interaction in Context Symposium*. IIiX '14. Regensburg, Germany: ACM, pp. 275–278. ISBN: 978-1-4503-2976-7. DOI: 10.1145/2637002.2637041. URL: http://doi.acm.org/10.1145/2637002.2637041 (cit. on pp. 43, 45).

Moreno-Torres, Jose G., Troy Raeder, RocíO Alaiz-RodríGuez, Nitesh V. Chawla, and Francisco Herrera (Jan. 2012). "A Unifying View on Dataset Shift in Classification." In: *Pattern Recogn.* 45.1, 521–530. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2011.06.019. URL:

https://doi.org/10.1016/j.patcog.2011.06.019 (cit. on p. 8).

Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press. ISBN: 0262018020, 9780262018029 (cit. on p. 49).

Newman-Griffis, Denis and Ayah Zirikly (July 2018). "Embedding Transfer for Low-Resource Medical Named Entity Recognition: A Case Study on Patient Mobility." In: *Proceedings of the BioNLP 2018 workshop*. Melbourne, Australia: Association for Computational Linguistics, pp. 1–11. DOI: 10.18653/v1/W18-2301. URL: https://www.aclweb.org/anthology/W18-2301 (cit. on pp. 59, 60, 62).

Nofer, Michael and Oliver Hinz (2015). "Using Twitter to Predict the Stock Market - Where is the Mood Effect?" In: *Business & Information Systems Engineering* 57.4, pp. 229–242 (cit. on pp. 43, 45).

Nopp, Clemens and Allan Hanbury (2015). "Detecting Risks in the Banking System by Sentiment Analysis." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 591–600. DOI: 10.18653/v1/D15-1071. URL: http://aclweb.org/anthology/D15-1071 (cit. on p. 45).

Pan, Sinno Jialin, Qiang Yang, Wei Fan, and Sinno Jialin Pan (ph. D (2010a). "A survey on transfer learning." In: *IEEE Transactions on Knowledge and Data Engineering* (cit. on pp. 18, 59).

Pan, Sinno Jialin, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen (2010b). "Cross-domain sentiment classification via spectral feature alignment." In: *Proceedings of the 19th international conference on World wide web*. ACM, pp. 751–760 (cit. on pp. 28, 59).

Pang, Bo and Lillian Lee (June 2005). "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales." In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 115–124. DOI: 10.3115/1219840.1219855. URL: https://www.aclweb.org/anthology/P05-1015 (cit. on p. 64).

Patel, Kevin, Divya Patel, Mansi Golakiya, Pushpak Bhattacharyya, and Nilesh Birari (Aug. 2017). "Adapting Pre-trained Word Embeddings For Use In Medical Coding." In: *BioNLP 2017*. Vancouver, Canada: Association for Computational Linguistics, pp. 302–306. DOI: 10.18653/v1/W17-2338. URL: https://www.aclweb.org/anthology/W17-2338 (cit. on p. 61).

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global Vectors for Word Representation." In: *EMNLP*. Vol. 14, pp. 1532–1543. URL: https://nlp.stanford.edu/pubs/glove.pdf (cit. on pp. 11, 57, 58, 60, 61).

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). "Deep

Contextualized Word Representations." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. Ed. by Marilyn A. Walker, Heng Ji, and Amanda Stent. Association for Computational Linguistics, pp. 2227–2237. DOI: 10.18653/v1/n18-1202. URL: https://doi.org/10.18653/v1/n18-1202 (cit. on pp. 14, 15).

Radford, Alec and Ilya Sutskever (2018). "Improving Language Understanding by Generative Pre-Training." In: (cit. on p. 15).

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). "Language Models are Unsupervised Multitask Learners." In: (cit. on p. 15).

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2019). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." In: *CoRR* abs/1910.10683. arXiv: 1910.10683. URL: http://arxiv.org/abs/1910.10683 (cit. on p. 15).

Raina, Rajat, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng (2007). "Self-taught learning: transfer learning from unlabeled data." In: *Proceedings of the 24th international conference on Machine learning*. Corvalis, Oregon: ACM, pp. 759–766. ISBN: 978-1-59593-793-3. DOI: 10.1145/1273496.1273592. URL: http://portal.acm.org/citation.cfm?id=1273592\&dl=GUIDE\&coll=GUIDE\&CFID=69344422\&CFTOKEN=94303924 (cit. on p. 59).

Rekabsaz, Navid, Mihai Lupu, Artem Baklanov, Alexander Dür, Linda Andersson, and Allan Hanbury (2017). "Volatility Prediction using Financial Disclosures Sentiments with Word Embedding-based IR Models." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1712–1721. DOI: 10.18653/v1/P17-1157. URL: https://doi.org/10.18653/v1/P17-1157 (cit. on p. 45).

Rietzler, Alexander, Sebastian Stabinger, Paul Opitz, and Stefan Engl (2020). "Adapt or Get Left Behind: Domain Adaptation through BERT Language Model Finetuning for Aspect-Target Sentiment Classification." In: *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*. Ed. by Nicoletta Calzolari et al. European Language Resources Association, pp. 4933–4941. URL: https://www.aclweb.org/anthology/2020.lrec-1.607/ (cit. on p. 17).

Roberts, Kirk (Dec. 2016). "Assessing the Corpus Size vs. Similarity Trade-off for Word Embeddings in Clinical NLP." In: *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 54–

63. URL: https://www.aclweb.org/anthology/W16-4208 (cit. on p. 62).

Rothe, Sascha, Sebastian Ebert, and Hinrich Schütze (2016). "Ultradense Word Embeddings by Orthogonal Transformation." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 767–777. DOI: 10.18653/v1/N16-1091. URL: http://aclweb.org/anthology/N16-1091 (cit. on p. 28).

Rothe, Sascha and Hinrich Schütze (2017). "AutoExtend: Combining Word Embeddings with Semantic Resources." In: *Computational Linguistics* 43.3, pp. 593–617. DOI: 10.1162/COLI_a_00294 (cit. on pp. 61, 62).

Ruder, Sebastian and Barbara Plank (2017). "Learning to select data for transfer learning with Bayesian Optimization." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Association for Computational Linguistics, pp. 372–382. DOI: 10.18653/v1/d17-1038. URL: https://doi.org/10.18653/v1/d17-1038 (cit. on p. 9).

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." In: *CoRR* abs/1910.01108. arXiv: 1910.01108. URL: http://arxiv.org/abs/1910.01108 (cit. on pp. 15, 17, 63, 65, 66).

Sarma, Prathusha Kameswara, Yingyu Liang, and William A. Sethares (2019). "Shallow Domain Adaptive Embeddings for Sentiment Analysis." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Association for Computational Linguistics, pp. 5548–5557. DOI: 10.18653/v1/D19-1557. URL: https://doi.org/10.18653/v1/D19-1557 (cit. on pp. 59–62, 64).

Sarma, Prathusha (June 2018). "Learning Word Embeddings for Data Sparse and Sentiment Rich Data Sets." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. New Orleans, Louisiana, USA: Association for Computational Linguistics, pp. 46–53. DOI: 10.18653/v1/N18-4007. URL: https://www.aclweb.org/anthology/N18-4007 (cit. on pp. 11, 57, 62, 64, 66).

Schnabel, Tobias and Hinrich Schütze (2014). "FLORS: Fast and Simple Domain Adaptation for Part-of-Speech Tagging." In: *Transactions of the Association for Computational Linguistics* 2, pp. 15–26.

DOI: 10.1162/tacl_a_00162. URL: https://www.aclweb.org/anthology/Q14-1002 (cit. on p. 59).

Sedinkina, Marina, Nikolas Breitkopf, and Hinrich Schütze (July 2019). "Automatic Domain Adaptation Outperforms Manual Domain Adaptation for Predicting Financial Outcomes." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 346–359. DOI: 10.18653/v1/P19-1034. URL: https://www.aclweb.org/anthology/P19-1034 (cit. on pp. 25–27, 31, 34–40).

Shah, Darsh, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov (2018). "Adversarial Domain Adaptation for Duplicate Question Detection." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1056–1063. DOI: 10.18653/v1/D18-1131. URL: https://www.aclweb.org/anthology/D18-1131 (cit. on p. 23).

Shen, Jian, Yanru Qu, Weinan Zhang, and Yong Yu (2018a). "Wasserstein Distance Guided Representation Learning for Domain Adaptation." In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, pp. 4058–4065. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17155 (cit. on p. 23).

Shen, Ying, Qiang Zhang, Jin Zhang, Jiyue Huang, Yuming Lu, and Kai Lei (2018b). "Improving Medical Short Text Classification with Semantic Expansion Using Word-Cluster Embedding." In: *Information Science and Applications 2018 - ICISA 2018, Hong Kong, China, June 25-27th, 2018*. Ed. by Kuinam J. Kim and Nakhoon Baek. Vol. 514. Lecture Notes in Electrical Engineering. Springer, pp. 401–411. DOI: 10.1007/978-981-13-1056-0_41. URL: https://doi.org/10.1007/978-981-13-1056-0_41 (cit. on p. 61).

Si, Jianfeng, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng (Jan. 2013). "Exploiting topic based twitter sentiment for stock prediction." English. In: *ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. Vol. 2. Association for Computational Linguistics (ACL), pp. 24–29. ISBN: 9781937284510 (cit. on p. 45).

Sridharan, Seshadri and Brian Murphy (2012). "Modeling Word Meaning: Distributional Semantics and the Corpus Quality-Quantity Trade-Off." In: *Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon*. Mumbai, India: The COLING 2012 Organizing Committee, pp. 53–68. URL: http://www.aclweb.org/anthology/W12-5105 (cit. on p. 1).

Takamura, Hiroya, Takashi Inui, and Manabu Okumura (2005). "Extracting Semantic Orientations of Words Using Spin Model." In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. ACL '05. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 133–140. DOI: 10.3115/1219840.1219857. URL: https://doi.org/10.3115/1219840.1219857 (cit. on p. 28).

Tang, Duyu, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu (2014). "Building Large-Scale Twitter-Specific Sentiment Lexicon : A Representation Learning Approach." In: *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*. Ed. by Jan Hajic and Junichi Tsujii. ACL, pp. 172–182. URL: https://www.aclweb.org/anthology/C14-1018/ (cit. on p. 28).

Tateisi, Yuka and Jun-ichi Tsujii (May 2004). "Part-of-Speech Annotation of Biology Research Abstracts." In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal: European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2004/pdf/528.pdf (cit. on p. 65).

Tetlock, Paul, Maytal Saar-Tsechansky, and Sofus Macskassy (Feb. 2008). "More Than Words: Quantifying Language to Measure Firms' Fundamentals." In: *Journal of Finance* 63, pp. 1437–1467. DOI: 10.2139/ssrn.923911 (cit. on p. 25).

Theil, Christoph Kilian, Sanja Stajner, and Heiner Stuckenschmidt (2018). "Word Embeddings-Based Uncertainty Detection in Financial Disclosures." In: *Proceedings of the First Workshop on Economics and Natural Language Processing*. Melbourne, Australia: Association for Computational Linguistics, pp. 32–37. URL: http://aclweb.org/anthology/W18-3104 (cit. on pp. 26, 27, 38).

Tsai, Ming-Feng and Chuan-Ju Wang (2014). "Financial Keyword Expansion via Continuous Word Vector Representations." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1453–1458. DOI: 10.3115/v1/D14-1152. URL: http://www.aclweb.org/anthology/D14-1152 (cit. on p. 27).

Tsvetkov, Yulia, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer (Aug. 2016). "Learning the Curriculum with Bayesian Optimization for Task-Specific Word Representation Learning." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 130–139. DOI: 10.18653/v1/P16-1013. URL: https://www.aclweb.org/anthology/P16-1013 (cit. on p. 9).

Turney, Peter D. (2002). "Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews."

In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 417–424. DOI: 10.3115/1073083.1073153. URL: https://doi.org/10.3115/1073083.1073153 (cit. on p. 27).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need." In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need (cit. on p. 15).

Velikovich, Leonid, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan T. McDonald (2010). "The viability of web-derived polarity lexicons." In: *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*. The Association for Computational Linguistics, pp. 777–785. URL: https://www.aclweb.org/anthology/N10-1119/ (cit. on p. 27).

Vicente, Iñaki San, Rodrigo Agerri, and German Rigau (2014). "Simple, Robust and (almost) Unsupervised Generation of Polarity Lexicons for Multiple Languages." In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*. Ed. by Gosse Bouma and Yannick Parmentier. The Association for Computer Linguistics, pp. 88–97. DOI: 10.3115/v1/e14-1010. URL: https://doi.org/10.3115/v1/e14-1010 (cit. on p. 28).

Vincent, P., H. Larochelle, Y. Bengio, and P.-A. Manzagol (2008). "Extracting and composing robust features with denoising autoencoders." In: *International Conference on Machine Learning proceedings* (cit. on pp. 20, 21).

Vo, Duy Tin and Yue Zhang (2016). "Don't Count, Predict! An Automatic Approach to Learning Sentiment Lexicons for Short Text." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 219–224. DOI: 10.18653/v1/P16-2036. URL: http://aclweb.org/anthology/P16-2036 (cit. on p. 28).

Wang, Baohua, Hejiao Huang, and Xiaolong Wang (2012). "A novel text mining approach to financial time series forecasting." In: *Neurocomputing* 83, pp. 136–145. DOI: 10.1016/j.neucom.2011.12.013. URL: https://doi.org/10.1016/j.neucom.2011.12.013 (cit. on p. 43).

Wang, Chuan-Ju, Ming-Feng Tsai, Tse Liu, and Chin-Ting Chang (2013). "Financial Sentiment Analysis for Risk Prediction." In: *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. Nagoya, Japan: Asian Federation of Natural Language Processing, pp. 802–808. URL: http://aclweb.org/anthology/I13-1097 (cit. on p. 45).

Wang, Leyi and Rui Xia (2017). "Sentiment Lexicon Construction with Representation Learning Based on Hierarchical Sentiment Supervision." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 502–510. DOI: 10.18653/v1/D17-1052. URL: http://aclweb.org/anthology/D17-1052 (cit. on p. 28).

Widdows, Dominic and Beate Dorow (2002). "A Graph Model for Unsupervised Lexical Acquisition." In: *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*. COLING '02. Taipei, Taiwan: Association for Computational Linguistics, pp. 1–7. DOI: 10.3115/1072228.1072342. URL: https://doi.org/10.3115/1072228.1072342 (cit. on p. 27).

Wu, Yonghui et al. (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation." In: *CoRR* abs/1609.08144. arXiv: 1609.08144. URL: http://arxiv.org/abs/1609.08144 (cit. on pp. 15, 16).

Xu, Hu, Bing Liu, Lei Shu, and Philip S. Yu (2019). "BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, pp. 2324–2335. DOI: 10.18653/v1/n19-1242. URL: https://doi.org/10.18653/v1/n19-1242 (cit. on p. 17).

Yang, Wei, Wei Lu, and Vincent W. Zheng (2017). "A Simple Regularization-based Algorithm for Learning Cross-Domain Word Embeddings." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Association for Computational Linguistics, pp. 2898–2904. DOI: 10.18653/v1/d17-1312. URL: https://doi.org/10.18653/v1/d17-1312 (cit. on pp. 12, 13, 61).

Yang, Yi and Jacob Eisenstein (June 2014). "Fast Easy Unsupervised Domain Adaptation with Marginalized Structured Dropout." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 538–544. DOI: 10.

3115/v1/P14-2088. URL: https://www.aclweb.org/anthology/P14-2088 (cit. on p. 22).

Yang, Yi and Jacob Eisenstein (2015). "Unsupervised Domain Adaptation with Feature Embeddings." In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.4385 (cit. on p. 59).

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le (2019). "XLNet: Generalized Autoregressive Pretraining for Language Understanding." In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, pp. 5754–5764. URL: http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding (cit. on pp. 15, 16).

Yijia, Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong lu (Dec. 2019). "BioWordVec, improving biomedical word embeddings with subword information and MeSH." In: *Scientific Data* 6. DOI: 10.1038/s41597-019-0055-0 (cit. on pp. 11, 61).

Yu, Jianfei, Minghui Qiu, Jing Jiang, Jun Huang, Shuangyong Song, Wei Chu, and Haiqing Chen (2018). "Modelling Domain Relationships for Transfer Learning on Retrieval-based Question Answering Systems in E-commerce." In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*. Ed. by Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek. ACM, pp. 682–690. DOI: 10.1145/3159652.3159685. URL: https://doi.org/10.1145/3159652.3159685 (cit. on p. 23).

Zhang, Jing, Wanqing Li, and Philip Ogunbona (2019). "Unsupervised domain adaptation: A multi-task learning-based method." In: *Knowl. Based Syst.* 186. DOI: 10.1016/j.knosys.2019.104975. URL: https://doi.org/10.1016/j.knosys.2019.104975 (cit. on p. 59).

Zhang, Zhu and Balaji Varadarajan (2006). "Utility scoring of product reviews." In: *Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006*. Ed. by Philip S. Yu, Vassilis J. Tsotras, Edward A. Fox, and Bing Liu. ACM, pp. 51–57. DOI: 10.1145/1183614.1183626. URL: https://doi.org/10.1145/1183614.1183626 (cit. on p. 46).