MAREIKE HARTMANN

TRANSFER LEARNING FOR COMPUTATIONAL CONTENT ANALYSIS

TRANSFER LEARNING FOR COMPUTATIONAL CONTENT ANALYSIS

MAREIKE HARTMANN

Ph.D.Thesis

October 2019

THESIS SUPERVISOR:
Anders Søgaard

ASSESSMENT COMMITTEE:
Christian Igel, University of Copenhagen
Katja Filippova, Google Research
David Schlangen, Bielefeld University

AFFILIATION:
Department of Computer Science
Faculty od Science
University of Copenhagen

Mareike Hartmann: *Transfer Learning for Computational Content Analysis*,, October 2019

ABSTRACT

Content analysis is a research technique that is concerned with the discovery of trends, patterns and differences in artifacts of human communication. It requires the reading and coding of data according to annotation guidelines, which is a labor-intensive process. In the times of mass communication, huge amounts of content are produced everyday. Analysing this content with respect to the social phenomena they capture is of interest to researchers in many fields. However, manual coding is impractical for such large amounts of data and automating the coding step could speed up the process significantly. Supervised machine learning is a promising approach in this direction, as such models can be applied to learn from human annotations and generalize to unseen data, making the coding of large amounts of content more feasible.

However, labeled data sets are expensive to generate. On the one hand, this leads to small training dataset sizes. On the other hand, it makes it valuable if a model can generalize across datasets from different domains and languages. Transfer learning is a machine learning method that enables such knowledge transfer between data from different distributions, leveraging as much data as possible and keeping the additional annotation efforts low.

This thesis investigates the use of transfer learning for automated content coding. In the first part of the work, we directly apply transfer learning to content coding tasks. We investigate how the methods can improve the task and show that transfer learning can overcome the problem of little training data by leveraging additional resources.

The second part of the work focuses on methods that enable knowledge transfer between languages. Such methods rely on word representations that capture meanings across languages. Unsupervised methods for learning such representations are attractive but unstable and we investigate the causes of these instabilities.

ABSTRACT IN DANISH – RESUMÉ PÅDANSK

Indholdsanalyse er en forskningsteknik, der beskæftiger sig med opdagelsen af tendenser, mønstre og forskelle i artefakter af menneskelig kommunikation. Det kræver læsning og kodning af data i henhold til retningslinjer for at markere data, som er en arbejdskrævende proces. I tiderne med massekommunikation produceres store mængder indhold hver dag. Analyse af dette indhold med hensyn til de sociale fænomener, de fanger, er af interesse for forskere på mange områder. Imidlertid er manuel kodning upraktisk for så store datamængder, og at automatisering af kodningstrinnet kan fremskynde processen markant. Overvåget maskinlæring er en lovende tilgang i denne retning, da sådanne modeller kan anvendes til at lære af menneskelige kommentarer og generalisere til usete data, hvilket gør kodningen af store mængder indhold mere gennemførlig.

Mærkede datasæt er imidlertid dyre at generere. På den ene side fører dette til små træningsdatastørrelser. På den anden side gør det det værdifuldt, hvis en model kan generalisere på tværs af datasæt fra forskellige domæner og sprog. Overførselslæring er en maskinlæringsmetode, der muliggør en sådan videnoverførsel mellem data fra forskellige distributioner, udnytter så mange data som muligt og holder de ekstra kommentarer indsats lave.

Denne afhandling undersøger brugen af overførselslæring til automatisk indholdskodning. I den første del af arbejdet anvender vi direkte overførselslæring til indholdskodningsopgaver. Vi undersøger, hvordan metoderne kan forbedre opgaven og viser, at overførselslæring kan løse problemet med lidt træningsdata ved at udnytte yderligere ressourcer.

Den anden del af arbejdet fokuserer på metoder, der muliggør videnoverførsel mellem sprog. Sådanne metoder er afhængige af ordrepræsentationer, der fanger betydninger på tværs af sprog. Ikkeovervågede metoder til at lære sådanne repræsentationer er attraktive, men ustabile, og vi undersøger årsagerne til disse ustabiliteter.

PUBLICATIONS

This is an article-based thesis. Chapters 3 to 7 each represent a peer-reviewed article. The articles are identical in content as they appear here and in the original publications, except for minor changes such as the correction of typos and the reformatting of tables and figures. The following articles are included in the thesis:

Hartmann, Mareike, and Anders Søgaard (2018). "Limitations of Cross-Lingual Learning from Image Search". In: *Proceedings of The Third Workshop on Representation Learning for NLP (Repl4NLP)*, pp.159–163

Hartmann, Mareike, Yova Kementchedjhieva and Anders Søgaard (2018). "Why is unsupervised alignment of English embeddings from different algorithms so hard". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.582–586

Hartmann, Mareike, Tallulah Jansen, Isabelle Augenstein and Anders Søgaard (2019). "Issue Framing in Online Discussion Fora". 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)*, pp. 1401-1407

Hartmann, Mareike, Yova Kementchedjhieva and Anders Søgaard (2019). "Comparing Unsupervised Word Translation Methods Step by Step". To appear in: *Proceedings of NeurIPS*

Hartmann, Mareike, Yevgeniy Golovchenko and Isabelle Augenstein (2019). "Mapping (Dis)information about the MH17 plane crash on Twitter". To appear in: *Proceedings of the 2nd Workshop on NLP for Internet Freedom (NLP4IF)*, pp. 1401-1407

I was also involved in the following publications that are not included in this thesis:

Kementchedjhieva, Yova, **Mareike Hartmann** and Anders Søgaard (2019). "Lost in Evaluation: Misleading Benchmarks for Bilingual Dictionary Induction". To appear in In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*

Pedersen, Bolette, Sanni Nimb, Anders Søgaard, **Mareike Hartmann** and Sussi Olsen (2018). "A Danish FrameNet Lexicon and an Annotated Corpus Used for Training and Evaluating a Semantic Frame Classifier". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*

Golovchenko, Yevgeniy, **Mareike Hartmann** and Rebecca Adler-Nissen (2018). "State, media and civil society in the information warfare over Ukraine: citizen curators of digital disinformation". In: *International Affairs*, vol. 94(5), Oxford University Press, pp. 975-994

ACKNOWLEDGMENTS

I want to say thank you to everyone that helped me in one way or the other to complete this Phd. My biggest thanks go to you, Anders, for your motivation, support, and unlimited optimism. I also want to thank you for gathering such a wonderful group of people in Copenhagen and for providing a fantastic work environment for us, I could not have wished for a nicer place for doing my Phd. Many thanks to my awesome colleagues, you are fantastic, and I enjoy spending time with you inside and outside the office a lot. Special thanks go to Maria, Yova, Desmond and Andreas for their valuable feedback and help with this thesis. Thank you Rebecca and Yev for your collaboration on the Digital Disinformation project. Thank you to all the wonderful people I met at LxMLS for all the fun we had, my flatmates for many dinners and board games, and my friends from back home for all their visits. Finally I want to thank my family, for encouraging me to go my way while always being by my side.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ML Machine Learning

MTL Multi-Task Learning

POS Part-of-Speech

NLP Natural Language Processing

MEN A dataset for multimodal distributional semantics

GAN Generative Adversarial Network

CNN Convolutional Neural Network

P@*k* Precision at *k*

MRR Mean Reciprocal Rank

KNN k nearest neighbor

VGG19 Very deep convolutional network for large-scale image recognition

LSTM Long-Short Term Memory Network

MUSE Multilingual Unsupervised or Supervised word Embeddings

CBOW Continuous Bag-of-Words

SVD Singular Value Decomposition

SGNS	Skipgram Negative Sampling
PMI	Pointwise Mutual Information
GloVe	Global Vectors for Word Representation
CSLS	Cross-domain Similarity Local Scaling
BDI	Bilingual Dictionary Induction
UBDI	Unsupervised Bilingual Dictionary Induction
GW	Gromov-Wasserstein
SGD	Stochastic Gradient Descent
ICP	Iterative Closest Point
MMD	Maximum Mean Discrepancy
DSB	Dutch Safety Board
JIT	Joint Investigation Committee
RNN	Recurrent Neural Network
SVM	Support Vector Machine
AUC	Area Under the Precision-Recall Curve
MaxEnt	Maximum Entropy
BUK	Buk surface to air missile system

Part I

BACKGROUND

1 INTRODUCTION

Communication is a key factor of human society, and hence the study of communication is a valuable tool to understand society (Lasswell, 1948). *Content analysis* is a methodology that is concerned with the discovery of trends, patterns and differences in artifacts of human communication, in order to infer or predict social phenomena. Communication artifacts can take various forms such as texts, speeches, images, television programs and melodies. A common aspect of all of these forms is that they were produced by someone (an author) in order to be consumed by someone (a reader). Hence, studying these artifacts allows researchers to make inferences about the contexts of their use. Content analysis is not only about understanding content, but also about using it to make inferences that can answer questions about society (Krippendorff, 2018).

For example, Budak and Watts, (2015) study if social movements actively shape the opinions and attitudes of their participants. In order to do so, they analyze tweets that were authored during the 2015 Gezi protests in Turkey, a series of protests that were started by environmentalists and turned into general protests against the government. From analysing the numbers of tweets that are supportive of Turkish opposition parties before, during and after the protests, they infer that solidarity among groups arises mainly from the fact that participants in the protests were generally more supportive of other groups beforehand, and less from the fact that different groups interact with each other during the protests.

Content analysis is usually applied to study phenomena that are not directly observable or accessible at the time the analysis is made. For example, in the Second World War, British researchers were able to predict major political and military campaigns by conducting content analysis of enemy propaganda (Guetzkow, 1959). In contrast to other research techniques used primarily in the social sciences, such as survey research, interviews, or social experiments, content analysis is non-invasive and non-reactive, i.e. humans producing or consuming the communication artifacts (authors or readers) are not influenced by the fact that their communication is subject to analysis (Salganik, 2019, Chapter 2.3.3). As communication is important to society, content analysis is a popular research technique in many fields of the social sciences to study phenomena of interest to these fields, such as mass communication research (Wimmer and Dominick, 2013), political science (Grimmer and Stewart, 2013), international relations research (Pashakhanlou, 2017), health research (Hsieh and Shannon,

2005), disaster research (Imran et al., 2013), psychology (Gondim and Bendassolli, 2014), and literary studies (Hoover, 2008).

The usual approach to content analysis involves several components, starting with the sampling of data and the breaking down of the data into units. These units are then subject to *content coding* or *content annotation*[1], i.e. they are assigned pre-defined categories or codes. The coding happens according to guidelines that the researcher developed beforehand, which are intended to make the coding process replicable. Finally, the researcher extrapolates from relations between these codes to phenomena that are not manifest in the data (Krippendorff, 2018, p.84). For example, for the above mentioned study on social movements, the researchers sampled Twitter data and annotated this data on the tweet level according to the authors' supportiveness of political parties and their participation in the movement. Then, they compare the distribution of these annotations for participants and non-participants before, during and after the protest, and make an inference about the influence of social movements on their participants.

1.0.1 CONTENT ANALYSIS IN THE DIGITAL AGE

Traditionally, the coding of content is carried out manually by domain experts, who read through considerable amounts of data and manually assign codes according to the coding guidelines. The coding of the content is the most time-consuming component of the analysis process (Wimmer and Dominick, 2013, p.171). This hurdle has become a major concern in the digital age of almost unlimited amounts of content being produced on a daily basis, and one of the main challenges of contemporary content analysis is the processing of such huge amounts of data (Krippendorff, 2018, p.5). Data of this size and variety have great potential to answer many questions about society, that could not be answered by looking at fewer data points (Salganik, 2019). However, content analysis for such large datasets and data from many different sources is infeasible if manually coded.

These considerations and the growing amount of content available in digital form led to a rise in adopting automated methods for content analysis. Early versions of so-called *computer-aided* content analysis involve dictionary-based coding systems (e.g. the General Inquirer (Stone, Dunphy, and Smith, 1966)), that automatically code content based on a word list specified by the researcher. Dictionary-based methods, however, come with drawbacks. Often, the categories to be coded are more nuanced and complex than what can be inferred based on the presence of single words or phrases, or the concepts these categories reflect are only implicit in the text. In these cases, such concepts can only be identified through complex combinations of sets of terms that are present in the text. Similarly, studies showed

1 We use the terms *coding*, *annotating* and *labeling* as synonyms throughout this thesis.

that the widely used method of using hashtags for content coding on Twitter is reliable for general topics, but fails to capture finer nuances such as the stance towards the topic (Budak and Watts, 2015). Hence, computational methods that go beyond the identification of pre-defined keywords are needed.

The computational extraction of complex patterns from data fall in the area of pattern recognition, and can be handled by Machine Learning (ML) methods. The ML methods applied for the computational analysis of text come from the field of Natural Language Processing (NLP). The application of NLP methods for content coding is attractive because it can easily be scaled to large amounts of data. Also, the content analysis process does not have to be changed to accomodate for the integration of the automated coding method, as NLP methods aim to identify codes at the same complexity level as the human coders (Scharkow, 2013; Wiedemann, 2016).

1.0.2 NATURAL LANGUAGE PROCESSING FOR CONTENT CODING

In the NLP community, methods for extracting useful information from text have been studied under the umbrella term of *text mining* (Hearst, 1999), and comprise a wide range of applications, such as text categorization (Sebastiani, 2002), event detection (Yang, Pierce, and Carbonell, 1998), hate speech detection (Schmidt and Wiegand, 2017), stance detection (Hanselowski et al., 2018), sentiment analysis (Pang, Lee, et al., 2008), and more. Here, dictionary-based approaches have been replaced by (supervised) ML methods, which have shown to be superior in capturing the semantics of text content for applications such as sentiment analysis (Bakliwal et al., 2013). Supervised methods for NLP rely on labeled training data on which model parameters are estimated, in order to then make inferences on unseen data. NLP classifiers for content coding as a component of content analysis have to be trained on hand-coded examples. Scharkow, (2013) notes that this is similar to the process of training a human coder, which is also trained how to code by looking at example documents.

Unfortunately, high quality hand-annotated data sets are expensive to generate, and especially so for content that is of interest for the social sciences and reflects complex phenomena that cannot be handled by dictionary-based approaches. This leads, on the one hand, to small annotated datasets for model training, which poses a challenge for the application of supervised ML models. On the other hand, the high cost of annotating data makes it more valuable to have models that can code new data sets without the need to manually label large amounts of new data. Hence, the goal for NLP models for content coding should be to *learn as much as possible from the available datasets, and be able to apply this knowledge to new datasets.*

For example, the Euromanifesto Study[2] (Braun et al., 2007) aims to collect and code manifestos issued to the elections for the European Parliament in all member countries of the European Union. Hence, data in new languages has to be coded as the number of member states increases. The same applies for the manifesto database[3] (Merz, Regel, and Lewandowski, 2016) that codes frames in political manifestos of democratic countries. Another example is the emergence of new forms of communication, such as social media, compared to traditional and long-studied newspaper media. For example, there is a are large dataset of newspaper articles coded with generic media frames that are applicable across platforms (Card et al., 2015), but only a much smaller dataset of tweets coded according to the same guidelines (Johnson, Jin, and Goldwasser, 2017). Except for our efforts in coding data from online discussion fora (Hartmann et al., 2019), there are no other datasets yet that extend this generic coding project to any other of the various digital media platforms. It would be incredibly valuable to be able to automatically code such new data in the same way the existing data was coded.

1.0.3 LEARNING FROM AVAILABLE DATA

NLP classifiers seem suited for reducing the manual effort in content coding, but the classifiers should be able to deal with small amounts of training data. These small dataset sizes result from the high cost of manual coding of content. For the same reason, the classifiers should be able to transfer their knowledge about relations between content and codes from one dataset to another. In the context of content analysis, these new datasets could for example comprise text data in a different language or artifacts from different communication channels, such as news articles compared to blog posts. In ML, such transfer can be accomplished using *transfer learning* (Pan and Yang, 2009).

The goal of transfer learning is to apply what a model has learned on one dataset to process a new dataset. For NLP classifiers applied to content analysis, this means that a model can learn how to assign codes to content from one dataset, and then transfer this knowledge to assign codes to a new dataset, according to the same paradigm that was used to code the old dataset. Transfer learning methods that have been explored for NLP and successfully been applied for text mining tasks include *cross-lingual learning* (learn from data in one language, apply it to data in another language) and *multi-task learning* (learn to solve several tasks simultaneously) and *domain adaptation* (learn from data in one domain, apply it to data in another domain) (Pan and Yang, 2009).

2 http://europeanelectionstudies.net/ees-study-components/euromanifesto-study

3 https://manifesto-project.wzb.eu/

From an ML perspective, transfer learning models help to overcome the lack of training data, and can improve model performance by exploiting information from complementary data (Pan and Yang, 2009). From an application-based perspective, transfer learning is interesting, because it allows the application of already existing models to new datasets with different text forms, e.g. texts from different platforms, or text in different languages, or even communication artifacts of different types such as images and text, without the need of manually annotating large portions of the new datasets.

Being able to apply the same model to text in different languages or from different platforms makes it possible to extract and combine information from multiple data sources, which is particularly interesting for content analysis across languages, or across communication channels. For example, Hanna, (2013) note that political activists from Bahrain tweet political messages in English rather than Arabic to avoid detection by state agents. Hence, analysing tweets in only one language paints an incomplete picture of the protests. On a similar note, Stewart, Pinter, and Eisenstein, (2018) found that people talk about independence in Catalan rather than Spanish. Almeida and Lichbach, (2003) note that media coverage is biased across platforms and that analysing content from as many data sources as possible should improve content analysis.

Hence, transfer learning seems a promising approach for enhancing computational content analysis. The present dissertation aims to explore this idea. Our central research question is:

Can transfer learning be useful for the automatic coding of content?

We approach the answer to this questions from two directions. First, we apply transfer learning for automatic coding of content and examine the use of such methods for the content coding task. Second, we focus on ways to improve transfer learning, in particular *cross-lingual* learning, and investigate how representations used for cross-lingual learning can be improved. Hence, the second research question for this thesis is:

How can we improve word representations that capture semantics across languages?

1.0.4 CONTRIBUTIONS OF THE THESIS

This thesis presents research into methods that enable automatic coding for text data from different domains or languages. The contributions that are presented in the following chapters are summarized below.

- We show to what extent NLP classifiers can be applied to label tweets for content analysis, finding that classifiers are most useful when applied as a pre-filter for human annotators (Chapter 3 in Part ii)
- We show that multi-task learning is beneficial for content coding in settings with no training data (Chapter 4 in Part ii)
- We introduce a new issue frame annotated corpus of online discussions (Chapter 4 in Part ii)
- We show that the induction of bilingual dictionaries from image data does not generalize to other parts of speech than nouns and identify reasons for this (Chapter 5 in Part iii)
- We show that training instabilities in a GAN-based architecture for aligning embeddings are likely caused by discriminator saddle points (Chapter 6 and 7 in Part iii)
- We provide a fair comparison between methods that align cross-lingual embeddings in an unsupervised setup (Chapter 7 in Part iii)

1.0.5 THESIS OVERVIEW

The thesis is divided in four parts. The remaining Chapter 2 in the first part of this thesis provides background on computational content analysis and transfer learning.

Part ii of the thesis focuses on the application of automated methods for content coding. In particular, we evaluate the use of transfer learning for overcoming the lack of training data by learning from additional data. In Chapter 3, we examine to what extent text classifiers can be used for content coding in the context of studying (dis)information flow on Twitter. We experiment with cross-lingual transfer based on aligned word embeddings and distant supervision, but find that performance does not improve, most likely because the additional data is noisy. Even though a neural classifier outperforms a hashtag-based baseline, the classifier does not generalize well beyond the training data. However, we find that the classifier can speed up the manual annotation process by pre-filtering data. In Chapter 4, we focus on knowledge transfer between domains, in particular news articles, tweets, and posts from online discussion fora. While frame-labeled

datasets exist for news articles and tweets, there is no training data available for online discussion fora. To alleviate this lack of training data, multi-task learning and domain adaptation are applied to learn from the datasets of news articles and tweets. We introduce a new data set of manually frame-labeled online discussions, which we use for evaluation.

Part iii of the thesis focuses on unsupervised methods that enable knowledge transfer between data in different languages, in particular the unsupervised induction of bilingual dictionaries, and the unsupervised alignment of cross-lingual word embeddings. In Chapter 5, we re-evaluate an approach for unsupervised bilingual dictionary induction on a larger dataset, that extends previous evaluations from nouns to verbs and adjectives. The approach uses image data associated with words as contexts for identifying word-level translations. The experiments reveal that the approach is promising for translating nouns, but does not generalize well to verbs and adjectives, most likely because those parts of speech refer to more abstract concepts.

Chapters 6 and 7 focus on methods for unsupervised alignment of word embeddings across languages. In Chapter 6, we analyse training instabilities of the MUSE system (Conneau et al., 2018), that performs unsupervised embedding alignment based on a generative adversarial network. We show that the system is unable to align two isomorphic graphs of English word embeddings learned with different embedding algorithms. We find that the system cannot navigate the highly non-convex loss landscape resulting from the different inductive biases of the embedding algorithms. The instabilities of the MUSE system are further investigated in Chapter 7, where we find indications that the training instability arises from saddle points in which the model gets stuck, and which cannot easily be overcome by varying hyperparameters such as batch size and learning rate. In the same Chapter, we present a fair comparison between several systems for unsupervised word embedding alignment, finding that the system suffers from instabilities, but has the highest potential to induce good initial seed dictionaries for subsequent iterative refinement.

Finally, Part iv summarizes and discusses the contributions made in the thesis.

2 BACKGROUND

The work presented in this thesis joins two methods that come from two different disciplines: content analysis, a method from the social sciences, and transfer learning, a method that comes from statistical learning. Hence, this background chapter has two parts. In the first part, we will provide background on automated content analysis and related work, including examples of content analyses that could profit from transfer learning. In the second part, we will give some background on transfer learning, with a focus on cross-lingual learning and a brief review of related work that successfully applies transfer learning for text mining tasks.

2.1 AUTOMATED METHODS FOR CONTENT ANALYSIS

A content analysis using automated methods follows the same workflow and is subject to the same requirements as traditional content analysis. This is visualized in Figure 2.1, which shows the workflow of a computational qualitative data analysis as seen by Wiedemann, (2016). We add the components of traditional content analysis as seen by Krippendorff on the left side of the Figure. The Figure shows that any of the components can be solved using automated methods, and each step can be evaluated using the metrics listed on the right. In the first step, relevant documents are selected. This step can be automated using information retrieval techniques (Manning, Raghavan, and Schütze, 2010). If a codebook needs to be defined, the data needs to be explored first. Unsupervised methods such as topic models (Blei and Lafferty, 2009) can be used to automate the exploration step.[1] Then, content is manually annotated according to the instructions in the codebook. If automated content coding is applied, the manual annotations serve to evaluate the quality of the automated annotations, or if supervised machine learning techniques are used for automated coding, also for model training. Finally, the annotated data is analysed. Automation can be integrated at various steps of the workflow.

In our work, we focus on the automation of the content coding step. Hence, the scope of our overview is limited to works that use automated methods for content coding.

1 If the data is annotated according to a pre-defined codebook, this step can be skipped

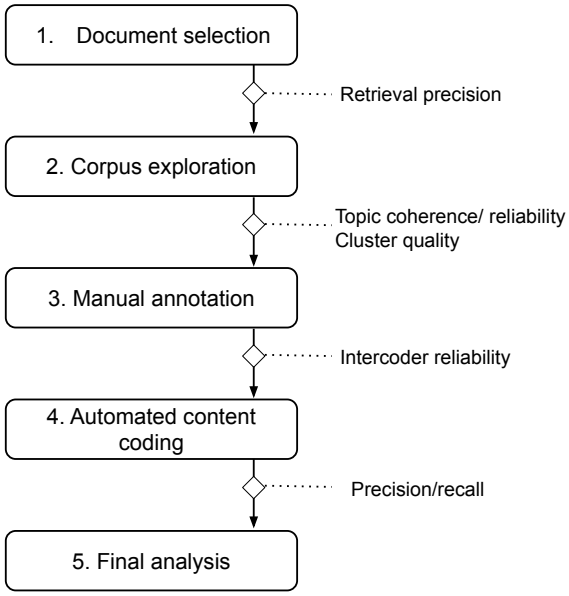

Figure 2.1: Workflow of computational qualitative data analysis as proposed by Wiedemann, (2016)(see p.227). We modify step 4 from *Active learning* to *Automated content coding* to reflect that any automated content coding method can be applied there.

2.1.1 AUTOMATED CONTENT CODING USING DICTIONARIES

In dictionary-based systems, researchers specify lists of words or symbols that they associate with categories of interest. The system then assigns these categories to content automatically according to the presence of dictionary entries in the text (Krippendorff, 2018; Wiedemann, 2016, Chapter 11.3, Chapter 2.3.1). The first dictionary-based systems were motivated by the idea that computers are more reliable in coding than humans, because they are more precise in identifying occurrences of words in text (Stone, Dunphy, and Smith, 1966). Dictionary-based coding with individual dictionaries was applied in a wide-range of studies, for example for coding sentiment in English political texts (Tumasjan et al., 2010; Young and Soroka, 2012) and the coding of events in English news paper articles for international relations research (King and Lowe, 2003; Schrodt, Davis, and Weddle, 1994).

LIMITATIONS Dictionary-based methods are only suited if the relevant content categories are manifest in the text, as they cannot capture more complex semantics arising from combinations of parts of the content. (Wiedemann, 2016, p.41). In most cases it is infeasible to generate exhaustive dictionaries that cover all potentially relevant words or expressions (Nelson et al., 2018, p.6).

Reducing the semantics of a text to the semantics of words without context is problematic. For example, Back, Küfner, and Egloff, (2010) draw incorrect inferences in their study about the emotional consequences of the September 11 terrorist acts in the US. The mistake is partly arising from the fact that their dictionary-based coding approach annotates an error message produced by a bot as *anger*, because it contains the keyword *critical* (see also Salganik, (2019, Chapter 2.3.9), Back, Küfner, and Egloff, (2011)). Another limitation of dictionary-based methods is that dictionaries are rarely generalized across projects. Dictionaries are expensive to generate, and in many cases project specific, i.e. they cannot be re-used across projects. Scharkow, 2012, p.79 found, that even more general dictionaries are not used in other works than the ones that introduced them.

2.1.2 AUTOMATED CONTENT CODING BASED ON META DATA

Coding based on meta data is similar to dictionary-based coding, where content is categorized based on the presence of meta data in the text, or by meta data that is associated with the content (for example when news articles are categorized according to the news outlet that produced them). One of the most popular instances of this approach is the coding of content from the micro blogging service Twitter [2] based on hashtags, a text meta tag inserted by the author. Hashtags were use in order to identify stance of the tweet author towards ISIS in a study on the causes of people supporting the group (Magdy, Darwish, and Weber, 2016), and author stance towards the Catalan independence referendum in Catalan and Spanish tweets (Stewart, Pinter, and Eisenstein, 2018). Conover et al., (2013) use hashtags to identify communication activities related to the Occupy Wall Street movement, an anti-capitalist protest movement in the US. González-Bailón, Borge-Holthoefer, and Moreno, (2013) use hashtags to examine how protest information in the Spanish outraged movement spreads on Twitter. Varol et al., (2014) study topical discussions about the Gezi movement in Turkey using lists of manually curated hashtags that are extended via bootstrapping.

LIMITATIONS Several studies suggest that care has to be taken when using hashtags for categorizing content. Budak and Watts, (2015) study the changes in attitude of participants of the Turkish Gezi movement using hashtags, and find that hashtags can only be used as proxy for support of the movement, if used by the opposition party. The majority of tweets that are authored by other parties and contain the target hashtag are hostile (Budak and Watts, 2015, p. 380). This shows that hashtags are not suited to identify fine-grained semantics of tweets. Another concern is that hashtag-based coding can only be

2 https://twitter.com/

applied to tweets that contain a hashtag, and excludes content without this feature from the study (Hanna, 2013, p.369) Finally, hashtag-based coding prevents a study from being extended across platforms that do not share this feature (Rogers, 2017, p.13).

2.1.3 CONTENT CODING USING TOPIC MODELS

Topic models are a popular family of unsupervised machine learning models among social scientists (Meeks and Weingart, 2012). The idea behind topic models is to model the generative process of a document collection assuming there is a fixed set of latent *topics*, i.e. distributions over words, that underlie the document generation (Blei and Lafferty, 2009). After approximating posterior distributions, each document in the collection can be described by a combination of topics that contributed to its generation. For automated content coding, a common practice is to label each data unit with the topic that is most prominent in the text (Nelson et al., 2018). Due to their unsupervised nature, topic models can also be applied for corpus exploration (Step 2 in Figure 2.1) prior to defining categories of interest.

Zhang, (2016) use topic models to track differences in topics of discussions before and after the umbrella protests in Hong Kong. Comparing the topic distributions over time between people that witnessed the protests and people that did not, they infer that physical witnesses are strongly impacted by the protests. Grimmer, (2010) use topic models to analyze press releases from US congress members with respect to how political actors portray themselves. Paul and Dredze, (2011) examine the use of Twitter for public health research using topic models. They correlate detected topics with public health metrics and find that Twitter contains much health related information that can be minded using topic models. Shen and Rose, (2019) use topic models to examine how Reddit users with different view points discuss a quarantine policy introduced by the platform. They find that right-leaning users frame the issue in terms of political censorship, while left-leaning users focus on issues surrounding the consistency in how the moderation is applied.

Topic models have also been applied in cross-lingual content analysis. Mimno et al., (2009) use multilingual topic modeling to track the development of topics across languages in proceedings of the European parliament. In a cross-national study, Sakamoto and Takikawa, (2017) compare the polarization in American and Japanese legislative speech.

LIMITATIONS Topic models have been criticized for instabilities arising from the randomness in approximating posterior distributions, i.e. they can produce different word to topic assignments for the same data, if ran with different random initializations (Koltcov, Koltsova,

and Nikolenko, 2014). The evaluation of topic models for content coding is difficult, as there are no guarantees that the model finds topics that coincide with categories assigned by humans (see e.g. Nelson et al., 2018, p.22–24).

2.1.4 CONTENT CODING USING SUPERVISED MACHINE LEARNING

The most commonly used supervised machine learning models applied for automatic content labeling are Support Vector Machine (SVM) (Boser, Guyon, and Vapnik, 1992; Cortes and Vapnik, 1995),Maximum Entropy (MaxEnt) classifiers, and Naive Bayes classifiers.

Nelson et al., (2018) benchmark three computer assisted approaches for content coding of news articles for the concept of economic inequality. They compare supervised ML methods (SVM and MaxEnt), a dictionary-based method and unsupervised ML methods (topic models and k-means clustering) based on their ability to reproduce the manually assigned codes. The supervised learning methods perform best, while the dictionary-based methods succeed in detecting explicit mentions of inequality and fail to identify latent mentions. The unsupervised methods fail to produce clusters that coincide with the manually assigned codes for most hyperparameters.

Hillard, Purpura, and Wilkerson, (2008) compare the performance of an SVM, a Naive Bayes classifier, and a Maximum Entropy classifier for the classification of congressional bills into up to 20 main topic and 226 sub-topic categories. They suggest to integrate only the high confidence classifier predictions into the content analysis, which are indicated by all three classifiers predicting the same category (see also Purpura and Hillard, (2006)).

Merz, Regel, and Lewandowski, (2016) use SVMs to code electoral manifestos on sentence level with one of 56 codes indicating the political framing. They report 42% precision, but point out that human agreement for the task is only 50%. They conclude that these results are promising for using the classifier in an semi-automated approach.

Several works use supervised machine learning for coding sentiment in political text, including political web-logs in English (Durant and Smith, 2006) and political tweets in Farsi (Vaziripour, Giraud-Carrier, and Zappala, 2016). Bakliwal et al., (2013) find that SVMs outperform dictionary-based baselines for sentiment classification in tweets. Johnson, Shukla, and Shukla, (2012) use a MaxEnt classifier to code the political sentiment of tweets and correlate it with survey data on the popularity of US president Obama. The results are mixed, as in the long-term the classifier shows negative correlation with the survey data.

Another approach is to use ensembles of classifiers for content coding. Stewart and Zhukov, (2009) study the public debate on the use of force in Russia. They use an ensemble of supervised classifiers on

bags-of-words to classify Russian government texts according to the expressed opinion as *Activist* or *Conservative*. Burscher et al., (2014) use an ensemble of classifiers on bags-of-words for detecting four generic frames in Dutch news articles. They examine if the trained models can generalize to news sources not included in the training set and find that this is possible with slight decreases in performance. They also find that the amount of improvement from increasing the amount of training examples varies between frames.

Morstatter et al., (2018) predict the degree of press freedom in a country by comparing the automatically assigned frames in government issued texts with t in news those in news paper articles. Their hypothesis is that the frames across both sets of texts will be more similar in countries with a lower degree of press freedom, as the government might dictate the frames the media outlets have to circulate. They perform sentence-level frame classification using several non-neural classifiers and an Long-Short Term Memory Network (LSTM) classifier.

Hopkins and King, (2010) introduce a supervised learning method that is based on the observation that social scientists are usually more interested in the proportion of data that falls into specific categories, than in accurate per data point classification. Even with increasing classification performance, the predictions can be biased with respect to category proportions. Hence they propose a method that directly estimates the proportions of categories in the unseen data from the annotated training data, without making individual classifications. This method is used by King, Pan, and Roberts, (2013) to study censorship in China. They classify censored/non-censored posts from a huge amount of Chinese social media sites and find that primarily posts that could lead to organized activity get censored, and not posts that criticize the government. Hanna, (2013) use Hopkins and King, (2010)'s method to study Egyptian online activism on Facebook. They look at tweets in Arabic and in Franco-Arabic and build language specific classifiers to automatically assign 5 categories reflecting the type of mobilization expressed in the tweet. Their reason for building separate classifiers is due to the bag-of-words approach not being able to capture cross-lingual similarities, which leads to small training set size (Hanna, 2013, p.380). They note that *Computer-aided content analysis methods also have not given sufficient attention to how to address multiple languages in a single corpus.* (p.384), a problem that we contribute to solve in this dissertation.

Field et al., (2018) perform a cross-lingual content analysis to study media manipulation strategies. They use cross-lingual word embeddings to project frames from a large manually annotated English data set to Russian news articles. In their analysis, they find that mentions of the US in the Russian newspaper increase in the month directly

following economic downturn in Russia, which they interpret as a distraction strategy.

LIMITATIONS The limitations of supervised machine learning methods for content coding lie in their (un)ability to model human behaviour (Grimmer and Stewart, 2013; Lake et al., 2017). The quality of the automated coding depends on the model's ability to learn from classifying the human annotations. This depends on many factors, such as the type of model and the difficulty of the task. Wiedemann, (2016) summarizes the factors that complicate the task of content coding compared to standard NLP text categorization (see p. 128 - 130). First, the categories of interest are often abstract concepts, which complicates the task. Second, the distribution over categories might be highly unbalanced, as the phenomena of interest might be rare. We encounter this problem in Chapter 3, where only 5% of the data correspond to the class of interest in a 3-way classification task. Finally, the models have to deal with small data set sizes for training, as data annotation is expensive and annotations might fit only a small research interest.

Building models that can deal with these challenges is likely to improve automated content coding using supervised models. The motivation behind using transfer learning for content coding is to alleviate the problem arising from the small amounts of training data.

2.2 TRANSFER LEARNING (FOR CONTENT CODING)

The idea behind transfer learning is to apply knowledge gained from solving one problem to solving another problem that is different but related (Pan and Yang, 2009; Ruder, 2019). This means the model is trained to solve one problem, and then applied or tested on another problem. As this setup violates the fundamental assumption that training data and testing data come from the same distribution, we cannot expect traditional ML models to perform well in this scenario. Transfer learning is a set of methods that enable the transfer of knowledge by accounting for the difference in source and target distribution (Pan and Yang, 2009). Two important concepts in transfer learning are *task* and *domain*, which we explain below following the definitions and notation of Ruder, (2019).

DOMAIN A domain D is defined as $D = \{\mathcal{X}, P(X)\}$. $\mathcal{X}$ is a feature space and $P(X)$ is a marginal probability distribution, where X is a training example represented by features in $\mathcal{X}$, i.e. $X = \{x_1, x_2, \cdots, x_n\} \in \mathcal{X}$.

In the context of content analysis, different domains can for example correspond to different communication channels, such as news articles

compared to social media platforms, or to different news outlets within one platform.

TASK A task T is defined as $T = \{\mathcal{Y}, P(Y), P(X|Y)\}$. $\mathcal{Y}$ is the set of possible labels , $P(Y)$ is a prior distribution over the labels, and $P(X|Y)$ is a conditional distribution that is learned from training examples. The training examples are pairs of feature vectors and corresponding labels, i.e. $\{x_i, y_i\}$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$.

In the context of content analysis, a task can for example correspond to a specific coding scheme. One task could be to label tweets according to polarity and another task to label these tweets according to authorship. In the first task, $\mathcal{Y}_\infty$ contains all possible polarity codes, while in the second task, $\mathcal{Y}_\in$ contains codes for all possible authors.

Transfer learning models learn from solving a *source* problem, i.e. a source task and a corresponding source domain, and transfer what they learned to a *target* problem, i.e a target task and corresponding target domain. Given a source domain D_s, a corresponding source task T_s, a target domain D_t and a corresponding target task T_t, the aim of transfer learning is to improve the learning of the conditional probability $P_t(Y_t|X_t)$ with knowledge from D_s and T_s, in cases where source and target domain are different ($D_s \neq D_t$), or source and target task are different ($T_s \neq T_t$).

The three types of transfer learning addressed in the later chapters of the present dissertation are *cross-lingual learning* (Chapters 3, 5, 6, 7), *domain adaptation* (Chapter 4), and *multi-task learning* (Chapter 4). According to the taxonomy introduced by Ruder, (2019) to classify transfer learning methods used in NLP, domain adaptation and cross-lingual learning are instances of transductive transfer learning. Transductive transfer learning is applied when source and target task are the same, and labeled data is only available in the source domain, which is different from the target domain. Consequently, knowledge has to be transferred from the source to the target domain. This method is referred to as domain adaptation. Cross-lingual learning is transductive transfer learning in cases where source and target domain are in different languages.

Multi-task learning is an instance of inductive transfer learning, as it leverages labeled data for both source and target task. Source and target task are different, and both tasks are learned simultaneously. Ideas from cross-lingual learning and domain adaptation can also be applied to enable transfer between languages and domains in a Multi-Task Learning (MTL) setup.

In the following, we provide some technical background on these methods using definitions and notation of Ruder et al., (2019). We also list examples of successful applications of these transfer learning methods for text mining tasks, in order to motivate their use for our content coding experiments. The choice of these methods for our

content coding tasks (see Part II) was motivated by the availability of data and the nature of the problems in the specific projects.

2.2.1 CROSS-LINGUAL LEARNING

Cross-lingual learning refers to knowledge transfer between source and target domain when texts in the two domains are in different languages. This is usually motivated by overcoming a lack of resources in the target language. For content analysis, this type of transfer can be useful for cross-national or cross-lingual studies, where content in two languages should be analysed simultaneously. It can also be useful to transfer annotations from text in one language to text in another language when manually labeled content only exists in the source language.

Early approaches to cross-lingual learning include the translation of text into a common language using machine translation systems (Balahur and Turchi, 2012; Bautin, Vijayarenu, and Skiena, 2008; Fortuna and Shawe-Taylor, 2005; Wan, 2008) and the projection of whole documents into a common multilingual space for applications such as information retrieval (Dumais, Landauer, and Littman, 1996).

In this thesis, we address two approaches to cross-lingual learning, *bilingual dictionary induction* and the *alignment of word embeddings*. Bilingual dictionary induction aims to induce word-level translations from monolingual corpora (Irvine and Callison-Burch, 2017), whereas the primary goal for word embedding alignment is to produce word representations that capture similarities across languages (Ruder, Vulić, and Søgaard, 2019). The two tasks are closely related, as the representations learned by word embedding alignment can be used to induce bilingual dictionaries, and Bilingual Dictionary Induction (BDI) is frequently used as an evaluation task for word embedding alignment. Both approaches enable cross-lingual transfer, as we will explain in the next sections.

2.2.1.1 BILINGUAL DICTIONARY INDUCTION

When large amounts of parallel data are available, word-to-word translations can be extracted from automatically induced word alignments between the parallel sentences (Brown et al., 1993). However, parallel data is expensive to generate and rarely available for low-resource languages. Hence, many BDI methods aim to induce dictionaries from non-parallel data. The idea behind those methods is based on the distributional hypothesis that words that occur in similar contexts have similar meanings (Harris, 1954). BDI approaches assume that this holds, even if the words and their contexts are from different languages. Hence, similar contexts across languages can serve as a signal to identify word translations. Several contextual signals that can be extracted from monolingual resources have been proposed,

such as co-occurrence patterns (Rapp, 1995), heterogeneity of contexts (Fung, 1995), the distribution of word usage over time (Schafer and Yarowsky, 2002), orthographic similarity (Melamed, 1995), or combinations thereof (Irvine and Callison-Burch, 2017). Other approaches, including our work presented in Chapter 5, explore the use of images as contexts to induce bilingual dictionaries (Bergsma and Goebel, 2011; Kiela, Vulic, and Clark, 2015; Vulić et al., 2016). There, the contexts of a word are computed based on the representations of images that are associated with the word.

To induce a bilingual dictionary, each source word w_s is represented by a vector representation of its context x_{w_s}. The translation(s) of w_s are identified by ranking the representations of all target words w_t according to their similarity to the source word. Similarity is typically computed as the cosine similarity $\text{sim}(x_{w_s}, x_{w_t}) = \frac{x_{w_s} \cdot x_{w_t}}{|x_{w_s}||x_{w_t}|}$ or variants derived thereof. The performance is evaluated using a rank-based metric such as Precision at k (P@k), i.e. the fraction of correct translations ranked among the k highest ranks averaged over all source words.

APPLICATIONS Inducing bilingual dictionaries can be useful for statistical machine translation, where the induced dictionaries are used to translate out-of-vocabulary words that the MT system did not encounter in the parallel training data(Irvine and Callison-Burch, 2017). Bilingual dictionaries can also directly be used to transfer knowledge across the language links in the dictionary. In this way, bilingual dictionaries have been used for cross-lingual transfer for metaphor detection in news articles (Tsvetkov et al., 2014), Named Entity Recognition (Zirikly and Hagiwara, 2015), dependency parsing (Durrett, Pauls, and Klein, 2012), cross-lingual information retrieval (Demner-Fushman and Oard, 2003; Grefenstette, 1998), the identification of subjective language in news articles (Mihalcea, Banea, and Wiebe, 2007), and multilingual sentiment analysis (Dashtipour et al., 2016). Bilingual dictionaries are also used as seeds to bootstrap cross-lingual embedding alignment.

2.2.1.2 CROSS-LINGUAL WORD EMBEDDINGS

Whereas BDI focusses on finding word-level translations, methods that learn cross-lingual word embeddings aim to learn word representations that capture the meaning of words across languages. Word embeddings, again, are based on the distributional hypothesis that contexts define the meaning of a word. Hence, it is possible to represent the meaning of a word in vector space based on the context it occurs in. Traditionally, this is done using *count-based* models, that represent each word by counting the co-occurrences with words in its context, and apply dimensionality reduction to the co-occurrence

matrix (Landauer, Foltz, and Laham, 1998; Levy, Goldberg, and Dagan, 2015). Recently, Mikolov et al., (2013a) introduced a highly efficient *predictive* model for learning distributed word representations. Here, word representations are learned by predicting a word's contexts in a supervised classification task[3]. During training, the representations are updated such that the probability of predicting the correct contexts gets maximized[4].

The similarities between monolingual word representations in vector space should reflect similarities between word meaning, e.g. representations of synonyms should lie close to each other. Cross-lingual word embeddings, analogously, capture the semantics of words within one language, but in addition they also capture the similarities of words across languages. For example, words that are translations of each other should have representations that lie close to each other in the vector space.

Such cross-lingual word representations can be used to solve the BDI task, or they can be used for cross-lingual transfer between models using directly, for example by using them in the task specific embedding layers of a multi-task setup (Lin et al., 2018).

There are numerous approaches to learning cross-lingual word embeddings (see e.g (Ruder, 2019, Table 3.3 on p.107)), that can be grouped into three categories. *Pseudo-multi-lingual corpora-based approaches* construct text data that contains words in both languages, and learn cross-lingual word embeddings by applying a standard model for computing monolingual word embeddings to that pseudo-multi-lingual corpus. *Joint methods* learn cross-lingual representations by optimizing an objective that aims at making similar words across languages similar. *Mapping-based approaches* map two sets of monolingual embeddings into a common space and are described in the next paragraph.

CROSS-LINGUAL WORD EMBEDDING ALIGNMENT Mapping based approaches aim to learn a mapping between two sets of monolingual word embeddings, such that words that are translations of each other have representations close to each other. The mapping is learned based on an initial bilingual *seed dictionary*, that captures correspondences between words and can be obtained in a supervised or an unsupervised setup (Ruder, Vulić, and Søgaard, 2019; Vulić et al., 2019). A mapping between two d-dimensional word embedding spaces is learned by aligning two matrices $S^{n \times d}$ and $T^{n \times d}$. The rows in S and T are ordered such that s_i and t_i correspond to translation pairs in the initial seed dictionary.

3 The supervision is given by observing a word's context in large otherwise unlabeled corpora.

4 It is shown that both types of models can be equivalent under certain conditions (Levy and Goldberg, 2014)

The mapping W is optimal when the mapped source word representations WS and the corresponding target word representations T are closest to each other, i.e. the optimal mapping is found as

$$W^* = \text{argmin}_W ||WS - T|| \tag{2.1}$$

A commonly adopted approach proposed by Artetxe, Labaka, and Agirre, (2017) is to apply equation (2.1) iteratively to bootstrap translation pairs for the seed dictionary, i.e. to start out with an initial *seed dictionary* and map the source words using the mapping W^* computed from that seed dictionary. Then, additional translation pairs are identified based on similarity between all mapped source representations and all target word representations. The new translation pairs are added to the seed dictionary and a new optimal mapping for the extended seed dictionary is computed.

UNSUPERVISED CROSS-LINGUAL WORD EMBEDDING ALIGNMENT

The most interesting question for word embedding alignment approaches is how to obtain the initial seed dictionary to compute S and T. Much work on cross-lingual learning is inspired by the idea that it can help transfer from resource-rich languages to low-resource languages for which few annotated data is available. Hence, much work is focusing on approaches that learn cross-lingual word embeddings with little to no supervision, i.e. the aim is to keep the initial seed dictionary as small and as cheap as possible. Cheap initial seed dictionaries have been generated by for example using numerals shared across languages (Artetxe, Labaka, and Agirre, 2018a), or identically spelled words (Smith et al., 2017; Søgaard, Ruder, and Vulić, 2018). *Unsupervised cross-lingual word embedding alignment* methods aim to learn alignments without *any* predefined initial seed dictionary, but solve equation (1) using a seed dictionary of word correspondences inferred in a completely unsupervised setup. The assumption of unsupervised approaches is that the structures of embedding spaces across languages are similar enough to enable alignment without initial hints about word correspondences. In other words, to allow unsupervised alignment, embeddings spaces have to be near isomorphic, i.e. the translations of neighbouring source word representations have to be neighbors in the target space as well (Barone, 2016; Søgaard, Ruder, and Vulić, 2018). Conneau et al., (2018) presented MUSE, the first unsupervised alignment system that could achieve results on par with supervised methods. Their system induces the initial seed dictionary using a GAN (Goodfellow et al., 2014). The network has two adversarial components, a generator that learns the mapping W^* and a discriminator that discriminates between mapped source representations and target representations. The generator learns the mapping such that the discriminator cannot discriminate between word representations from source and target language, which leads to a mapping that makes the mapped source and target representations as similar as possible.

Søgaard, Ruder, and Vulić, (2018) found that the isomorphism condition does not hold for many language pairs and that the GAN training is instable and highly dependent on parameter configurations such as training domain and language pairs. In Chapter 6 we present a study that investigates the causes of these instabilities, where we find that they are due to a problem in the interplay between generator and discriminator. Many other unsupervised methods have been proposed following the MUSE system, that apply different strategies to matching source and target embedding spaces without supervision. Many of them are inspired by overcoming the robustness issues of MUSE. These systems are described in more detail in Chapter 7, where we present a step by step comparison between several unsupervised models, that focuses on equal conditions for automatically inducing the initial seed dictionary.

2.2.1.3 EVALUATION

Analogously to monolingual word embeddings, the quality of cross-lingual word embeddings can be evaluated using multi-lingual word similarity datasets (Ruder, Vulić, and Søgaard, 2019). Here, the cross-lingual quality of the embeddings is determined through correlation between model-assigned and human-assigned similarity scores between pairs of source and target words. However, using this method for evaluation is questionable (Ruder, Vulić, and Søgaard, 2019), as for the monolingual case it was shown that human judgment in this task can be subjective (Faruqui et al., 2016).

Another common approach is to use BDI as evaluation task. Here, translation pairs are identified using a retrieval function on the cross-lingual word representations. The gold dictionaries used most frequently for evaluation of cross-lingual embedding alignments (Conneau et al., 2018; Dinu, Lazaridou, and Baroni, 2015) cover a wide range of languages but were generated automatically. Hence, they contain considerable amounts of noise. In Kementchedjhieva, Hartmann, and Søgaard, (2019), we analyse the shortcomings of the datasets released along with the MUSE system and show how the quality of dictionaries can distort differences in model performances (see Appendix **??**). Besides the quality of resources, further concerns about BDI as evaluation task were raised by Glavas et al., (2019), who found there is little correlation between BDI performance and performance in downstream tasks such as document classification, information retrieval, and natural language inference.

2.2.1.4 APPLICATIONS

Besides improving core-NLP tasks such as dependency parsing (Guo et al., 2015; Søgaard et al., 2015) and part-of-speech tagging (Gouws and Søgaard, 2015), cross-lingual embeddings have also proven useful

for applications relevant to content coding, including document classification (Klementiev, Titov, and Bhattarai, 2012; Søgaard et al., 2015), the detection of textual similarity (Glavaš et al., 2018) and sentiment analysis (Mogadala and Rettinger, 2016).

2.2.2 MULTI-TASK LEARNING

The idea behind MTL is that by learning different tasks simultaneously, the models can combine information from all the tasks, which can be beneficial to solve the individual tasks. (Caruana, 1993). It was introduced in the context of NLP by Collobert et al., (2011).

Ruder, (2019) summarizes five reasons for why multi-task learning might improve performance over single-task learning. First, it implicitly performs data augmentation by increasing the sample size. Second, it can help to focus the model's attention on relevant features that generalize across tasks. Third, it enable eavesdropping, i.e. getting information that is needed for solving a task by looking at other tasks. Fourth, it encourages learning models that generalize across tasks. Fifth, solving multiple tasks at once acts as a regularizer preventing the individual models from overfitting.

2.2.2.1 PARAMETER SHARING

The most straight-forward approach to MTL is *hard parameter sharing*. In the context of neural models, hard parameter sharing means that per task, one neural network is trained, but the hidden layers are shared between all the tasks. The output layers are task-specific (see left part of Figure 4.1 on page 50). Other approaches to parameter sharing do not share any layers between tasks, but the parameters of each network are constrained to be close to each other. Such approaches are referred to as *soft parameter sharing* (Ruder, 2019, p.49).

2.2.2.2 AUXILIARY TASKS

Multi-task learning can be applied when different tasks ($T_s \neq T_t$) need to be solved simultaneously. Even if we are interested in only the target task, it might be beneficial to simultaneously solve the source task(s).[5] In that case, the source tasks (here referred to as *auxiliary tasks*) should be chosen such that they are most helpful to solve the target task (Alonso and Plank, 2016; Bingel and Søgaard, 2017; Bjerva, 2017).

The most straightforward choice for an auxiliary task is a supervised classification task related to the target task (Ruder, 2019). In the case of text classification, this could be text classification at different granularity levels (Balikas, Moura, and Amini, 2017) or a different text classification task (Liu, Qiu, and Huang, 2017; Zhang et al., 2017a).

5 The number of tasks that are solved simultaneously is not limited to two.

2.2.2.3 DOMAIN ADAPTATION WITH AN ADVERSARIAL AUXILIARY TASK

If MTL involves data from different domains, it can be beneficial to have an auxiliary task that explicitly aims to overcome this difference between domains. A popular strategy is to use an *adversarial* auxiliary task. Similarly to the adversarial component in GANs for embedding alignment, the adversarial auxiliary task makes the model learn representations that are similar across domains. Ganin and Lempitsky, (2015) introduced a setup for domain adaptation, in which a model is optimized to perform as poorly as possible in discriminating between source and target domain. This is achieved by optimizing the model for a binary classification task of discriminating between domains, but reversing the direction of the loss gradient in the backpropagation step. Due to the gradient reversal, the model learns representations that discriminate between the domains as little as possible.

Even though this task is supervised, the supervision comes from discriminating between domains, and hence otherwise unlabeled data can be used for this task.

2.2.2.4 APPLICATIONS

MTL has successfully been applied for many text mining tasks that are relevant for content coding, including the classification of product reviews (Liu, Qiu, and Huang, 2017), sentiment analysis in product reviews (Wu and Huang, 2016), joint detection of sentiment and topics in tweets (Huang et al., 2013), author stance in tweets (Ma, Gao, and Wong, 2018), and hate speech detection on Twitter (Waseem, Thorne, and Bingel, 2018).

In Chapter 4, we apply multi-task learning with hard parameter sharing and an adversarial auxiliary task in order to assign frame labels from news articles and tweets to posts from online discussion fora.

Part II

TRANSFER LEARNING FOR AUTOMATED CONTENT CODING

3

MAPPING (DIS-)INFORMATION FLOW ABOUT THE MH17 PLANE CRASH

ABSTRACT

Digital media enables not only fast sharing of information, but also disinformation. One prominent case of an event leading to circulation of disinformation on social media is the MH17 plane crash. Studies analysing the spread of information about this event on Twitter have focused on small, manually annotated datasets, or used proxys for data annotation. In this work, we examine to what extent text classifiers can be used to label data for subsequent content analysis, in particular we focus on predicting pro-Russian and pro-Ukrainian Twitter content related to the MH17 plane crash. Even though we find that a neural classifier improves over a hashtag based baseline, labeling pro-Russian and pro-Ukrainian content with high precision remains a challenging problem. We provide an error analysis underlining the difficulty of the task and identify factors that might help improve classification in future work. Finally, we show how the classifier can facilitate the annotation task for human annotators.

3.1 INTRODUCTION

Digital media enables fast sharing of information, including various forms of false or deceptive information. Hence, besides bringing the obvious advantage of broadening information access for everyone, digital media can also be misused for campaigns that spread disinformation about specific events, or campaigns that are targeted at specific individuals or governments. Disinformation, in this case, refers to intentionally misleading content (Fallis, 2015).

A prominent case of a disinformation campaign are the efforts of the Russian government to control information during the Russia-Ukraine crisis (Pomerantsev and Weiss, 2014). One of the most important events during the crisis was the crash of Malaysian Airlines (MH17) flight on July 17, 2014. The plane crashed on its way from Amsterdam to Kuala Lumpur over Ukrainian territory, causing the death of 298 civilians. The event immediately led to the circulation of competing narratives about who was responsible for the crash (see Section 3.2), with the two most prominent narratives being that the plane was either shot down by the Ukrainian military, or by Russian separatists in Ukraine supported by the Russian government (Oates, 2016). The latter theory was confirmed by findings of an international investigation team. In

this work, information that opposes these findings by promoting other theories about the crash is considered disinformation. When studying disinformation, however, it is important to acknowledge that our fact checkers (in this case the international investigation team) may be wrong, which is why we focus on both of the narratives in our study.

MH17 is a highly important case in the context of international relations, because the tragedy has not only increased Western, political pressure against Russia, but may also continue putting the government's global image at stake. In 2020, at least four individuals connected to the Russian separatist movement will face murder charges for their involvement in the MH17 crash (Harding, 2019), which is why one can expect the waves of disinformation about MH17 to continue spreading. The purpose of this work is to develop an approach that may help both practitioners and scholars of political science, international relations and political communication to detect and measure the scope of MH17-related disinformation.

Several studies analyse the framing of the crash and the spread of (dis)information about the event in terms of pro-Russian or pro-Ukrainian framing. These studies analyse information based on manually labeled content, such as television transcripts (Oates, 2016) or tweets (Golovchenko, Hartmann, and Adler-Nissen, 2018; Hjorth and Adler-Nissen, 2019). Restricting the analysis to manually labeled content ensures a high quality of annotations, but prohibits analysis from being extended to the full amount of available data. Another widely used method for classifying misleading content is to use distant annotations, for example to classify a tweet based on the domain of a URL that is shared by the tweet, or a hashtag that is contained in the tweet (Gallacher et al., 2018; Grinberg et al., 2019; Guess, Nagler, and Tucker, 2019). Often, this approach treats content from uncredible sources as misleading (e.g. misinformation, disinformation or fake news). This methods enables researchers to scale up the number of observations without having to evaluate the fact value of each piece of content from low-quality sources. However, the approach fails to address an important issue: Not all content from uncredible sources is necessarily misleading or false and not all content from credible sources is true. As often emphasized in the propaganda literature, established media outlets too are vulnerable to state-driven disinformation campaigns, even if they are regarded as credible sources (Chomsky and Herman, 1988; Jowett and O'donnell, 2014; Taylor, 2003)[1].

In order to scale annotations that go beyond metadata to larger datasets, NLP models can be used to automatically label text content. For example, several works developed classifiers for annotating text content with frame labels that can subsequently be used for large-scale

1 The U.S. media coverage of weapons of mass destruction in Iraq stands as one of the most prominent examples of how generally credible sources can be exploited by state authorities.

content analysis (**Card2015**; Boydstun et al., 2014; Field et al., 2018; Hartmann et al., 2019; Ji and Smith, 2017; Johnson, Jin, and Goldwasser, 2017; Naderi and Hirst, 2017; Tsur, Calacci, and Lazer, 2015). Similarly, automatically labeling attitudes expressed in text (Augenstein et al., 2016; Hasan and Ng, 2013; Walker et al., 2012b; Zubiaga et al., 2018) can aid the analysis of disinformation and misinformation spread (Zubiaga et al., 2016). In this work, we examine to which extent such classifiers can be used to detect pro-Russian framing related to the MH17 crash, and to which extent classifier predictions can be relied on for analysing information flow on Twitter.

MH17 RELATED (DIS-)INFORMATION FLOW ON TWITTER We focus our classification efforts on a Twitter dataset introduced in Golovchenko, Hartmann, and Adler-Nissen, (2018), that was collected to investigate the flow of MH17-related information on Twitter, focusing on the question who is distributing (dis-)information. In their analysis, the authors found that citizens are active distributors, which contradicts the widely adopted view that the information campaign is only driven by the state and that citizens do not have an active role. To arrive at this conclusion, the authors manually labeled a subset of the tweets in the dataset with pro-Russian/pro-Ukrainian frames and build a retweet network, which has Twitter users as nodes and edges between two nodes if a retweet occurred between the two associated users. An edge was considered as *polarized* (either pro-Russian or pro-Ukrainian), if at least one retweet between the two users connected by the edge was pro-Russian/pro-Ukrainian. Then, the amount of polarized edges between users with different profiles (e.g. citizen, journalist, state organ) was computed.

Labeling more data via automatic classification (or computer-assisted annotation) of tweets could serve an analysis as the one presented in Golovchenko, Hartmann, and Adler-Nissen, (2018) in two ways. First, more edges could be labeled.[2] Second, edges could be labeled with higher precision, i.e. by taking more tweets comprised by the edge into account. For example, one could decide to only label an edge as polarized if at least half of the retweets between the users were pro-Ukrainian/pro-Russian.

CONTRIBUTIONS We evaluate different classifiers that predict frames for unlabeled tweets in Golovchenko, Hartmann, and Adler-Nissen, (2018)'s dataset, in order to increase the number of polarized edges in the retweet network derived from the data. This is challenging due to a skewed data distribution and the small amount of training data for the pro-Russian class. We try to combat the data sparsity using a data augmentation approach, but have to report a negative result as we

2 Only 26% of the available tweets in Golovchenko, Hartmann, and Adler-Nissen, (2018)'s dataset are manually labeled.

find that data augmentation in this particular case does not improve classification results. While our best neural classifier clearly outperforms a hashtag-based baseline, generating high quality predictions for the pro-Russian class is difficult: In order to make predictions at a precision level of 80%, recall has to be decreased to 23%. Finally, we examine the applicability of the classifier for finding new polarized edges in a retweet network and show how, with manual filtering, the number of pro-Russian edges can be increased by 29%. We make our code, trained models and predictions publicly available[3].

3.2 COMPETING NARRATIVES ABOUT THE MH17 CRASH

We briefly summarize the timeline around the crash of MH17 and some of the dominant narratives present in the dataset. On July 17, 2014, the MH17 flight crashed over Donetsk Oblast in Ukraine. The region was at that time part of an armed conflict between pro-Russian separatists and the Ukrainian military, one of the unrests following the Ukrainian revolution and the annexation of Crimea by the Russian government. The territory in which the plane fell down was controlled by pro-Russian separatists.

Right after the crash, two main narratives were propagated: Western media claimed that the plane was shot down by pro-Russian separatists, whereas the Russian government claimed that the Ukrainian military was responsible. Two organisations were tasked with investigating the causes of the crash, the Dutch Safety Board (DSB) and the Dutch-led Joint Investigation Committee (JIT). Their final reports were released in October 2015 and September 2016, respectively, and conclude that the plane had been shot down by a missile launched by a Buk surface to air missile system (BUK). The BUK was stationed in an area controlled by pro-Russian separatists when the missile was launched, and had been transported there from Russia and returned to Russia after the incident. These findings are denied by the Russian government until now. There are several other crash-related reports that are frequently mentioned throughout the dataset. One is a report by Almaz-Antey, the Russian company that manufactured the BUK, which rejects the DSB findings based on mismatch of technical evidence. Several reports backing up the Dutch findings were released by the investigative journalism website Bellingcat.[4]

The crash also sparked the circulation of several alternative theories, many of them promoted in Russian media (Oates, 2016), e.g. that the plane was downed by Ukrainian SU25 military jets, that the plane attack was meant to hit Putin's plane that was allegedly traveling the same route earlier that day, and that the bodies found in the plane had already been dead before the crash.

3 https://github.com/coastalcph/mh17

4 https://www.bellingcat.com/

3.3 DATASET

For our classification experiments, we use the MH17 Twitter dataset introduced by Golovchenko, Hartmann, and Adler-Nissen, (2018), a dataset collected in order to study the flow of (dis)information about the MH17 plane crash on Twitter. It contains tweets collected based on keyword search[5] that were posted between July 17, 2014 (the day of the plane crash) and December 9, 2016.

Golovchenko, Hartmann, and Adler-Nissen, (2018) provide annotations for a subset of the English tweets contained in the dataset. A tweet is annotated with one of three classes that indicate the framing of the tweet with respect to responsibility for the plane crash. A tweet can either be *pro-Russian* (Ukrainian authorities, NATO or EU countries are explicitly or implicitly held responsible, or the tweet states that Russia is not responsible), *pro-Ukrainian* (the Russian Federation or Russian separatists in Ukraine are explicitly or implicitly held responsible, or the tweet states that Ukraine is not responsible) or *neutral* (neither Ukraine nor Russia or any others are blamed). Example tweets for each category can be found in Table 3.2. These examples illustrate that the framing annotations do not reflect general polarity, but polarity with respect to responsibility to the crash. For example, even though the last example in the table is in general pro-Ukrainian, as it displays the separatists in a bad light, the tweet does not focus on responsibility for the crash. Hence the it is labeled as neutral.

Table 3.1 shows the label distribution of the annotated portion of the data as well as the total amount of original tweets, and original tweets plus their retweets/duplicates in the network. A *retweet* is a repost of another user's original tweet, indicated by a specific syntax (RT @username:). We consider as *duplicate* a tweet with text that is identical to an original tweet after preprocessing (see Section 3.5.1). For our classification experiments, we exclusively consider original tweets, but model predictions can then be propagated to retweets and duplicates.

3.4 CLASSIFICATION MODELS

For our classification experiments, we compare three classifiers, a hashtag-based baseline, a logistic regression classifier and a Convolutional Neural Network (CNN).

HASHTAG-BASED BASELINE Hashtags are often used as a means to assess the content of a tweet (Dhingra et al., 2016; Efron, 2010; Godin

5 These keywords were: MH17, Malazijskij [and] Boeing (in Russian), #MH17, #Pray4MH17, #PrayforMH17. The dataset was collected using the Twitter *Garden hose*, which means that it contains a 10% of all tweets within the specified period that matched the search criterion.

	Label	Original	All
Labeled	Pro-Russian	512	4,829
	Pro-Ukrainian	910	12,343
	Neutral	6,923	118,196
Unlabeled	-	192,003	377,679
Total	-	200,348	513,047

Table 3.1: Label distribution and dataset sizes. Tweets are considered *original* if their preprocessed text is unique. *All* tweets comprise original tweets, retweets and duplicates.

et al., 2013). We identify hashtags indicative of a class in the annotated dataset using the pointwise mutual information (pmi) between a hashtag *hs* and a class *c*, which is defined as

$$\text{pmi}(hs, c) = \log \frac{\text{p}(hs, c)}{\text{p}(hs)\ \text{p}(c)} \tag{3.1}$$

We then predict the class for unseen tweets as the class that has the highest pmi score for the hashtags contained in the tweet. Tweets without hashtag (5% of the tweets in the development set) or with multiple hashtags leading to conflicting predictions (5% of the tweets in the development set) are labeled randomly. We refer to to this baseline as HS_PMI.

LOGISTIC REGRESSION CLASSIFIER As non-neural baseline we use a logistic regression model.[6] We compute input representations for tweets as the average over pre-trained word embedding vectors for all words in the tweet. We use fasttext embeddings (Bojanowski et al., 2017) that were pre-trained on Wikipedia.[7]

CONVOLUTIONAL NEURAL NETWORK CLASSIFIER As neural classification model, we use a CNN (Kim, 2014), which has previously shown good results for tweet classification (Dhingra et al., 2016; Santos and Gatti, 2014).[8] The model performs 1d convolutions over a sequence of word embeddings. We use the same pre-trained fasttext embeddings as for the logistic regression model. We use a model with one convolutional layer and a relu activation function, and one max pooling layer. The number of filters is 100 and the filter size is set to 4.

6 As non-neural alternative, we also experimented with SVMs. These showed inferior performance to the regression model.

7 In particular, with cross-lingual experiments in mind (see Section 3.7), we used embeddings that are pre-aligned between languages available here https://fasttext.cc/docs/en/aligned-vectors.html

8 We also ran intitial experiments with recurrent neural networks (RNNs), but found that results were comparable with those achieved by the CNN architecture, which runs considerably faster.

Label	Example tweet
Pro-Ukrainian	Video - Missile that downed MH17 'was brought in from Russia' @peterlane5news
	RT @mashable: Ukraine: Audio recordings show pro-Russian rebels tried to hide #MH17 black boxes.
	Russia Calls For New Probe Into MH17 Crash. Russia needs to say, ok we fucked up.. Rather than play games
	@IamMH17 STOP LYING! You have ZERO PROOF to falsely blame UKR for #MH17 atrocity. You will need to apologize.
Pro-Russian	Why the USA and Ukraine, NOT Russia, were probably behind the shooting down of flight #MH17
	RT @Bayard_1967: UKRAINE Eyewitness Confirm Military Jet Flew Besides MH17 Airliner: BBC ...
	RT @GrahamWP_UK: Just read through #MH17 @bellingcat report, what to say - written by frauds, believed by the gullible. Just that.
Neutral	#PrayForMH17 :(
	RT @deserto_fox: Russian terrorist stole wedding ring from dead passenger #MH17

Table 3.2: Example tweets for each of the three classes.

3.5 EXPERIMENTAL SETUP

We evaluate the classification models using 10-fold cross validation, i.e. we produce 10 different datasplits by randomly sampling 60% of the data for training, 20% for development and 20% for testing. For each fold, we train each of the models described in Section 3.4 on the training set and measure performance on the test set. For the CNN and LogReg models, we upsample the training examples such that each class has as many instances as the largest class (Neutral). The final reported scores are averages over the 10 splits.[9]

3.5.1 TWEET PREPROCESSING

Before embedding the tweets, we replace urls, retweet syntax (RT @user_name:) and @mentions (@user_name) by placeholders. We lowercase all text and tokenize sentences using the StandfordNLP pipeline (Qi et al., 2018). If a tweet contains multiple sentences, these are concatenated. Finally, we remove all tokens that contain non-

9 We train with the same hyperparameters on all splits, these hyperparameters were chosen according to the best macro f score averaged over 3 runs with different random seeds on *one* of the splits.

Model	Macro-avg F1	Macro-avg AUC	Pro-Russian F1	Pro-Russian AUC	Pro-Ukrainian F1	Pro-Ukrainian AUC	Neutral F1	Neutral AUC
RANDOM	0.25	-	0.10	-	0.16	-	0.47	-
HS_PMI	0.25	-	0.10	-	0.16	-	0.48	-
LOGREG	0.59	0.53	0.38	0.34	0.51	0.41	0.88	0.86
CNN	**0.69**	**0.71**	**0.55**	**0.57**	**0.59**	**0.60**	**0.93**	**0.94**

Table 3.3: Classification results on the English MH17 dataset measured as F1 and AUC.

alphanumeric symbols (except for dashes and hashtags) and strip the hashtags from each token, in order to increase the number of words that are represented by a pre-trained word embedding.

3.5.2 EVALUATION METRICS

We report performance as F1-scores, which is the harmonic mean between precision and recall. As the class distribution is highly skewed and we are mainly interested in accurately classifying the classes with low support (pro-Russian and pro-Ukrainian), we report macro-averages over the classes. In addition to F1-scores, we report the Area Under the Precision-Recall Curve (AUC).[10] We compute an AUC score for each class by converting the classification task into a one-vs-all classification task.

3.6 RESULTS

The results of our classification experiments are presented in Table 3.3. Figure 3.1 shows the per-class precision-recall curves for the LOGREG and CNN models as well as the confusion matrices between classes.[11]

COMPARISON BETWEEN MODELS We observe that the hashtag baseline performs poorly and does not improve over the random baseline. The CNN classifier outperforms the baselines as well as the LOGREG model. It shows the highest improvement over the LOGREG for the pro-Russian class. Looking at the confusion matrices, we observe that for the LOGREG model, the fraction of True Positives is equal between the pro-Russian and the pro-Ukrainian class. The CNN model produces a higher amount of correct predictions for the pro-Ukrainian than for the pro-Russian class. The absolute number of pro-Russian

10 The AUC is computed according to the trapezoidal rule, as implemented in the sklearn package (Pedregosa et al., 2011)

11 Both the precision-recall curves and the confusion matrices were computed by concatenating the test sets of all 10 datasplits

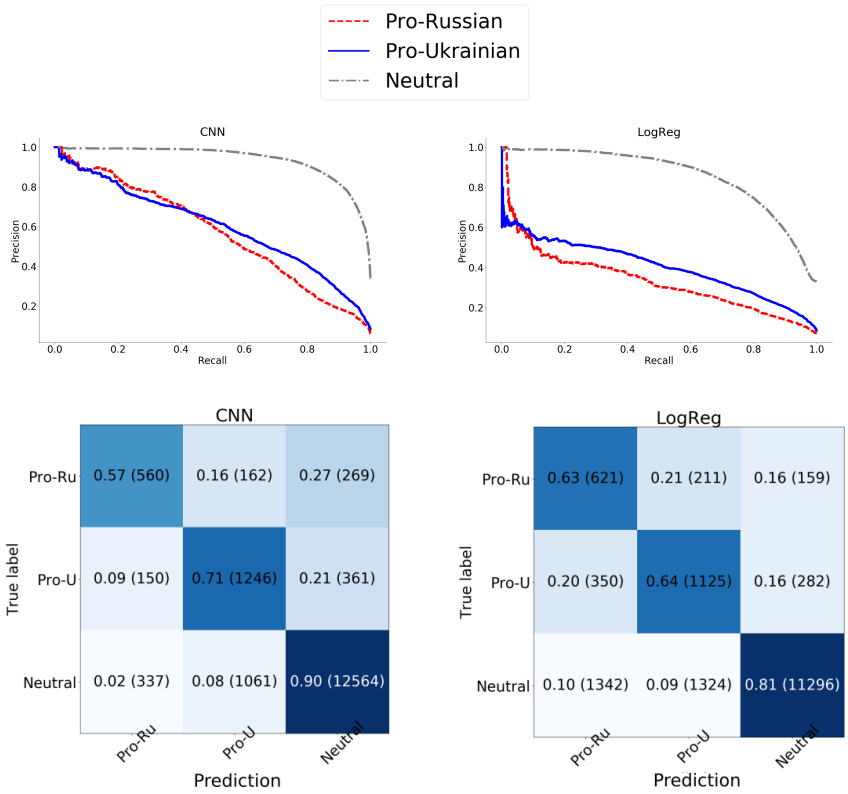

Figure 3.1: Confusion matrices for the CNN (left) and the logistic regression model (right). The y-axis shows the true label while the x-axis shows the model prediction.

True Positives is lower for the CNN, but so is in return the amount of misclassifications between the pro-Russian and pro-Ukrainian class.

PER-CLASS PERFORMANCE With respect to the per class performance, we observe a similar trend across models, which is that the models perform best for the neutral class, whereas performance is lower for the pro-Ukrainian and pro-Russian classes. All models perform worst on the pro-Russian class, which might be due to the fact that it is the class with the fewest instances in the dataset.

Considering these results, we conclude that the CNN is the best performing model and also the classifier that best serves our goals, as we want to produce accurate predictions for the pro-Russian and pro-Ukrainian class without confusing between them. Even though the CNN can improve over the other models, the classification performance for the pro-Russian and pro-Ukrainian class is rather low. One obvious reason for this might be the small amount of training data, in particular for the pro-Russian class.

In the following, we briefly report a negative result on an attempt to combat the data sparseness with cross-lingual transfer. We then perform an error analysis on the CNN classifications to shed light on the difficulties of the task.

3.7 DATA AUGMENTATION EXPERIMENTS USING CROSS-LINGUAL TRANSFER

The annotations in the MH17 dataset are highly imbalanced, with as few as 512 annotated examples for the pro-Russian class. As the annotated examples were sampled from the dataset at random, we assume that there are only few tweets with pro-Russian stance in the dataset. This observation is in line with studies that showed that the amount of disinformation on Twitter is in fact small (Grinberg et al., 2019; Guess, Nagler, and Tucker, 2019). In order to find more pro-Russian training examples, we turn to a resource that we expect to contain large amounts of pro-Russian (dis)information. The *Elections integrity dataset*[12] was released by Twitter in 2018 and contains the tweets and account information for 3,841 accounts that are believed to be Russian trolls financed by the Russian government. While most tweets posted after late 2014 are in English language and focus on topics around the US elections, the earlier tweets in the dataset are primarily in Russian language and focus on the Ukraine crisis (Howard et al., 2018). One feature of the dataset observed by Howard et al., (2018) is that several hashtags show high peakedness (Kelly et al., 2012), i.e. they are posted with high frequency but only during short intervals, while others are persistent during time.

We find two hashtags in the Elections integrity dataset with high peakedness that were exclusively posted within 2 days after the MH17 crash and that seem to be pro-Russian in the context of responsibility for the MH17 crash: #КиевСкажиПравду (*Kiew tell the truth*) and #Киевсбилбоинг (*Kiew made the plane go down*). We collect all tweets with these two hashtags, resulting in 9,809 Russian tweets that we try to use as additional training data for the pro-Russian class in the MH17 dataset. We experiment with cross-lingual transfer by embedding tweets via aligned English and Russian word embeddings.[13] However, so far results for the cross-lingual models do not improve over the CNN model trained on only English data. This might be due to the fact that the additional Russian tweets rather contain a general pro-Russian frame than specifically talking about the crash, but needs further investigation.

Error cat.	True class	Model prediction	id	Tweet
			a)	RT @ChadPergram: Hill intel sources say Russia has the capability to potentially shoot down a #MH17 but not Ukraine.
	Pro-U	Pro-R		

12 https://about.twitter.com/en_us/values/elections-integrity.html#data

13 We use two sets of monolingual fasttext embeddings trained on Wikipedia (Bojanowski et al., 2017) that were aligned relying on a seed lexicon of 5000 words via the RCSLS method (Joulin et al., 2018)

			b)	RT @C4ADS: .@bellingcat's new report says #Russia used fake evidence for #MH17 case to blame #Ukraine URL
			c)	The international investigation blames Russia for MH17 crash URL #KievReporter #MH17 #Russia #terror #Ukraine #news #war
		Pro-R	Pro-U	d)	RT @RT_com: BREAKING: No evidence of direct Russian link to #MH17 - US URL URL
				e)	RT @truthhonour: Yes Washington was behind Eukraine jets that shot down MH17 as pretext to conflict with Russia. No secrets there
				f)	Ukraine Media Falsely Claim Dutch Prosecutors Accuse Russia of Downing MH17: Dutch prosecutors de URL #MH17 #alert
				g)	@Werteverwalter @Ian56789 @ClarkeMicah no SU-25 re #MH17 believer has ever been able to explain it,facts always get in their way
		Pro-U	Pro-R	h)	Rebel theories on #MH17 "total nonsense", Ukrainian Amb to U.S. Olexander Motsyk interviewed by @jaketapper via @cnn
				i)	Ukrainian Pres. says it's false "@cnnbrk: Russia says records indicate Ukrainian warplane was flying within 5 km of #MH17 on day of crash.
II				j)	Russia has released some solid evidence to contradict @EliotHiggins + @bellingcat's #MH17 report. http://t.co/3leYfSoLJ3
		Pro-R	Pro-U	k)	RT @masamikuramoto: @MJoyce2244 The jets were seen by Russian military radar and Ukrainian eyewitnesses. #MH17 @Fossibilities @irina
		Pro-R	Pro-U	l)	RT @katehodal: Pro-Russia separatist says #MH17 bodies "weren't fresh" when found in Ukraine field,suggesting already dead b4takeoff
				m)	RT @NinaByzantina: #MH17 redux: 1) #Kolomoisky admits involvement URL 2) gets $1.8B of #Ukraine's bailout funds

III	Pro-U	Pro-R	n)	#Russia again claiming that #MH17 was shot down by air-to-air missile, which of course wasn't russian-made. #LOL URL
			o)	RT @20committee: New Moscow line is #MH17 was shot down by a Ukrainian fighter. With an LGBT pilot, no doubt.
	Pro-R	Pro-U	p)	RT @merahza: If you believe the pro Russia rebels shot #MH17 then you'll believe Justine Bieber is the next US President and that Coke is a
			q)	So what @AC360 is implying is that #US imposed sanctions on #Russia, so in turn they shot down a #Malaysia jet carrying #Dutch people? #MH17
			r)	RT @GrahamWP_UK: #MH17 1. A man on sofa watching YouTube thinks it was a 'separatist BUK'. 2. Man on site for over 25 hours doesn't.

Table 3.4: Examples for the different error categories. Error category I are cases where the correct class can easily be inferred from the text. For error category II, the correct class can be inferred from the text with event-specific knowledge. For error category III, it is necessary to resolve humour/satire in order to infer the intended meaning that the speaker wants to communicate.

3.8 ERROR ANALYSIS

In order to integrate automatically labeled examples into a network analysis that studies the flow of polarized information in the network, we need to produce high precision predictions for the pro-Russian and the pro-Ukrainian class. Polarized tweets that are incorrectly classified as neutral will hurt an analysis much less than neutral tweets that are erroneously classified as pro-Russian or pro-Ukrainian. However, the worst type of confusion is between the pro-Russian and pro-Ukrainian class. In order to gain insights into why these confusions happen, we manually inspect incorrectly predicted examples that are confused between the pro-Russian and pro-Ukrainian class. We analyse the misclassifications in the development set of all 10 runs, which results in 73 False Positives of pro-Ukrainian tweets being classified as pro-Russian (referred to as *pro-Russian False Positives*), and 88 False Positives of pro-Russian tweets being classified as pro-

Ukrainian (referred to as *pro-Ukrainian False Positives*). We can identify three main cases for which the model produces an error:

1. the correct class can be directly inferred from the text content easily, even without background knowledge
2. the correct class can be inferred from the text content, given that event-specific knowledge is provided
3. the correct class can be inferred from the text content if the text is interpreted correctly

For the pro-Russian False Positives, we find that 42% of the errors are category I and II errors, respectively, and 15% of category III. For the pro-Ukrainian False Positives, we find 48% category I errors, 33% category II errors and and 13% category III errors. Table 3.4 presents examples for each of the error categories in both sets which we will discuss in the following.

CATEGORY I ERRORS Category I errors could easily be classified by humans following the annotation guidelines (see Section 3.3). One difficulty can be seen in example f). Even though no background knowledge is needed to interpret the content, interpretation is difficult because of the convoluted syntax of the tweet. For the other examples it is unclear why the model would have difficulties with classifying them.

CATEGORY II ERRORS Category II errors can only be classified with event-specific background knowledge. Examples g), i) and k) relate to the theory that a Ukrainian SU25 fighter jet shot down the plane in air. Correct interpretation of these tweets depends on knowledge about the SU25 fighter jet. In order to correctly interpret example j) as pro-Russian, it has to be known that the bellingcat report is pro-Ukrainian. Example l) relates to the theory that the shoot down was a false flag operation run by Western countries and the bodies in the plane were already dead before the crash. In order to correctly interpret example m), the identity of *Kolomoisky* has to be known. He is an anti-separatist Ukrainian billionaire, hence his involvement points to the Ukrainian government being responsible for the crash.

CATEGORY III ERRORS Category III errors occur for examples that can only be classified by correctly interpreting the tweet authors' intention. Interpretation is difficult due to phenomena such as irony as in examples n) and o). While the irony is indicated in example n) through the use of the hashtag *#LOL*, there is no explicit indication in example o).
Interpretation of example q) is conditioned on world knowledge as well as the understanding of the speakers beliefs. Example r) is pro-Russian as it questions the validity of the assumption AC360 is making,

but we only know that because we know that the assumption is absurd. Example s) requires to evaluate that the speaker thinks people on site are trusted more than people at home.

From the error analysis, we conclude that category I errors need further investigation, as here the model makes mistakes on seemingly easy instances. This might be due to the model not being able to correctly represent Twitter specific language or unknown words, such as *Eukraine* in example e). Category II and III errors are harder to avoid and could be improved by applying reasoning (Wang and Cohen, 2015) or irony detection methods (Van Hee, Lefever, and Hoste, 2018).

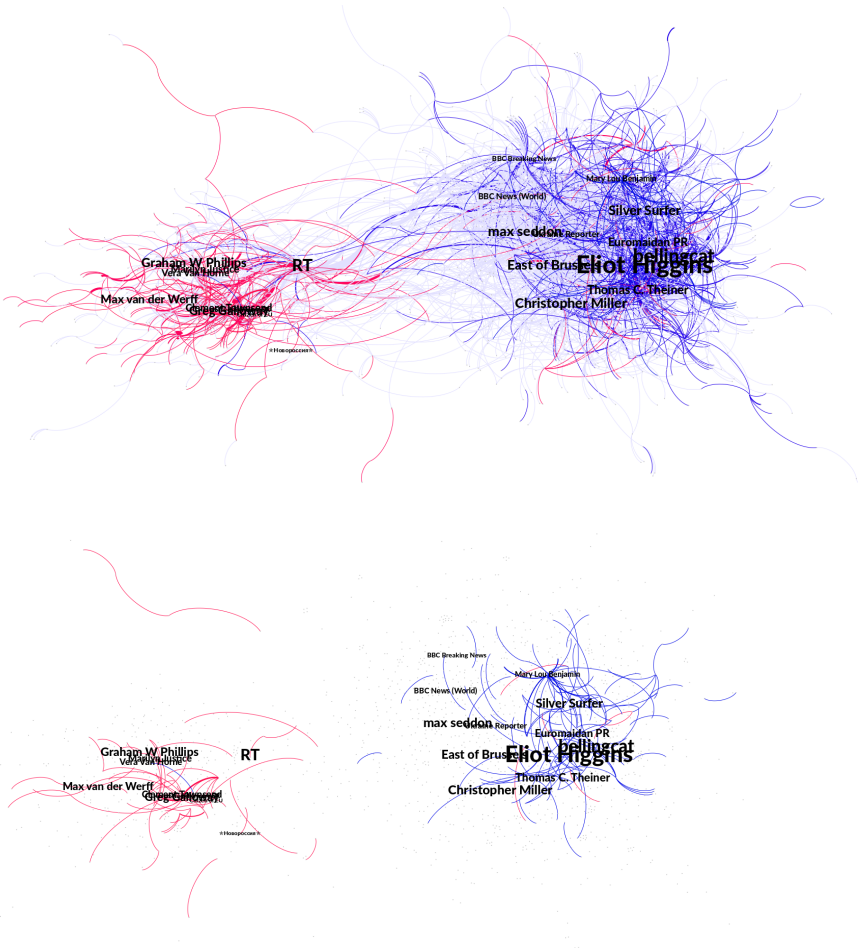

Figure 3.2: The upper plot shows the original k10 retweet network as computed by Golovchenko, Hartmann, and Adler-Nissen, (2018) together with the new edges that were added after manually re-annotating the classifier predictions. The bottom plot only visualizes the new edges that we could add by filtering the classifier predictions. Pro-Russian edges are colored in red, pro-Ukrainian edges are colored in dark blue and neutral edges are colored in grey. Both plots were made using The Force Atlas 2 layout in gephi (Bastian, Heymann, and Jacomy, 2009).

3.9 INTEGRATING AUTOMATIC PREDICTIONS INTO THE RETWEET NETWORK

Finally, we apply the CNN classifier to label new edges in Golovchenko, Hartmann, and Adler-Nissen, (2018)'s retweet network, which is shown in Figure 3.2. The retweet network is a graph that contains users as nodes and an edge between two users if the users are retweeting each other.[14] In order to track the flow of polarized information, Golovchenko, Hartmann, and Adler-Nissen, (2018) label an edge as polarized if at least one tweet contained in the edge was manually annotated as pro-Russian or pro-Ukrainian. While the network shows a clear polarization, only a small subset of the edges present in the network are labeled (see Table 3.5).

Automatic polarity prediction of tweets can help the analysis in two ways. Either, we can label a previously unlabeled edge, or we can verify/confirm the manual labeling of an edge, by labeling additional tweets that are comprised in the edge.

3.9.1 PREDICTING POLARIZED EDGES

In order to get high precision predictions for unlabeled tweets, we choose the probability thresholds for predicting a pro-Russian or pro-Ukrainian tweet such that the classifier would achieve 80% precision on the test splits (recall at this precision level is 23%). Table 3.5 shows the amount of polarized edges we can predict at this precision level. Upon manual inspection, we however find that the quality of predictions is lower than estimated. Hence, we manually re-annotate the pro-Russian and pro-Ukrainian predictions according to the official annotation guidelines used by Golovchenko, Hartmann, and Adler-Nissen, 2018. This way, we can label 77 new pro-Russian edges by looking at 415 tweets, which means that 19% of the candidates are hits. For the pro-Ukrainian class, we can label 110 new edges by looking at 611 tweets (18% hits). Hence even though the quality of the classifier predictions is too low to be integrated into the network analysis right away, the classifier drastically facilitates the annotation process for human annotators compared to annotating unfiltered tweets (from the original labels we infer that for unfiltered tweets, only 6% are hits for the pro-Russian class, and 11% for the pro-Ukrainian class).

14 Golovchenko, Hartmann, and Adler-Nissen, (2018) use the k10 core of the network, which is the maximal subset of nodes and edges, such that all included nodes are connected to at least k other nodes (Seidman, 1983), i.e. all users in the network have interacted with at least 10 other users.

	Pro-R	Pro-U	Neutral	Total
# labeled edges in k10	270	678	2193	3141
# candidate edges	349	488	-	873
# added after filtering predictions	77	**110**	-	187

Table 3.5: Number of labeled edges in the k10 network before and after augmentation with predicted labels. Candidates are previously unlabeled edges for which the model makes a confident prediction. The total number of edges in the network is 24,602.

3.10 CONCLUSION

In this work, we investigated the usefulness of text classifiers to detect pro-Russian and pro-Ukrainian framing in tweets related to the MH17 crash, and to which extent classifier predictions can be relied on for producing high quality annotations. From our classification experiments, we conclude that the real-world applicability of text classifiers for labeling polarized tweets in a retweet network is restricted to prefiltering tweets for manual annotation. However, if used as a filter, the classifier can significantly speed up the annotation process, making large-scale content analysis more feasible.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their helpful comments. The research was carried out as part of the 'Digital Disinformation' project, which was directed by Rebecca Adler-Nissen and funded by the Carlsberg Foundation (project number CF16-0012).

4

ISSUE FRAMING IN ONLINE DISCUSSION FORA

ABSTRACT

In online discussion fora, speakers often make arguments for or against something, say birth control, by highlighting certain aspects of the topic. In social science, this is referred to as *issue framing*. In this paper, we introduce a new issue frame annotated corpus of online discussions. We explore to what extent models trained to detect issue frames in newswire and social media can be transferred to the domain of discussion fora, using a combination of multi-task and adversarial training, assuming only unlabeled training data in the target domain.

4.1 INTRODUCTION

The *framing* of an issue refers to a choice of perspective, often motivated by an attempt to influence its perception and interpretation (Chong and Druckman, 2007; Entman, 1993). The way issues are framed can change the evolution of policy as well as public opinion (Dardis et al., 2008; Iyengar, 1991). As an illustration, contrast the statement *Illegal workers depress wages* with *This country is abusing and terrorizing undocumented immigrant workers*. The first statement puts focus on the economic consequences of immigration, whereas the second one evokes a morality frame by pointing out the inhumane conditions under which immigrants may have to work. Being exposed to primarily one of those perspectives might affect the public's attitude towards immigration.

Computational methods for frame classification have previously been studied in news articles (Card et al., 2015) and social media posts (Johnson, Jin, and Goldwasser, 2017). In this work, we introduce a new benchmark dataset, based on a subset of the 15 generic frames in the *Policy Frames Codebook* by (Boydstun et al., 2014). We focus on frame classification in *online discussion fora*, which have become crucial platforms for public dialogue on social and political issues. Table 1 shows example annotations, compared to previous annotations for news articles and social media. Dialogue data is substantially different from news articles and social media, and we therefore explore ways to transfer information from these domains, using multi-task and adversarial learning, providing non-trivial baselines for future work in this area.

Platform: Online discussions
Economic Frame, Topic: Same sex marriage `But as we have seen, supporting same-sex marriage saves money.`
Legality Frame, Topic: Same sex marriage `So you admit that it is a right and it is being denied?`
Platform: News articles
Economic Frame, Topic: Immigration `Study Finds That Immigrants Are Central to Long Island Economy`
Legality Frame, Topic: Same sex marriage `Last week, the Iowa Supreme Court granted same-sex couples the right to marry.`
Platform: Twitter
Legality Frame, Topic: Same sex marriage `Congress must fight to ensure LGBT people have the full protection of the law everywhere in America. #EqualityAct`

Table 4.1: Example instances from the datasets described in §4.2 and 4.3.

CONTRIBUTIONS We present a new issue-frame annotated dataset that is used to evaluate issue frame classification in online discussion fora. Issue frame classification was previously limited to news and social media. As manual annotation is expensive, we explore ways to overcome the lack of labeled training data in the target domain with multi-task and adversarial learning, leading to improved results in the target domain.[1]

RELATED WORK Previous work on automatic frame classification focused on news articles and social media. Card et al., (2016) predict frames in news articles at the document level, using clusters of latent dimensions and word-based features in a logistic regression model. Ji and Smith, (2017) improve on previous work integrating discourse structure into a recursive neural network. Naderi and Hirst, (2017) use the same resource, but make predictions at the sentence level, using topic models and recurrent neural networks. Johnson, Jin, and Goldwasser, (2017) predict frames in social media data at the micro-

1 Code and annotations are available at https://github.com/coastalcph/issue_framing.

Frames	1	13	5	6	7
# instances	78	96	234	166	186

Table 4.2: Class distribution in the online discussion test set. The frame labels correspond to the classes *Economic* (1), *Political* (13), *Legality, Jurisprudence and Constitutionality* (5), *Policy prescription and evaluation* (6) and *Crime and Punishment* (7).

post level, using probabilistic soft logic based on lists of keywords, as well as temporal similarity and network structure. All the work mentioned above uses the generic frames of Boydstun et al., (2014)'s Policy Frames Codebook. Baumer et al., (2015) predict words perceived as frame-evoking in political news articles with hand-crafted features. Field et al., (2018) analyse how Russian news articles frame the U.S. using a keyword-based cross-lingual projection setup. Tsur, Calacci, and Lazer, (2015) use topic models to analyze issue ownership and framing in public statements released by the US congress. Besides work on frame classification, there has recently been a lot of work on aspects closely related to framing, such as subjectivity detection (Lin, He, and Everson, 2011), detection of biased language (Recasens, Danescu-Niculescu-Mizil, and Jurafsky, 2013) and stance detection (Augenstein et al., 2016; Ferreira and Vlachos, 2016; Mohammad et al., 2016).

Model	Task	Domain	Labelset	# classes	# sequences
Baseline	Main task	News articles	Frames	5	10,480
	Target task	Online disc. (test)	Frames	5	692
Multitask	+Aux task	Tweets	Frames	5	1,636
	+Aux task	Online disc.	Argument quality	2	3,785
Adversarial	+Adv task	Online disc. + News articles	Domain	2	4,731 + 10,480
		Online disc. (dev)	Frames	5	176

Table 4.3: Overview over the data and labelsets for the different tasks. The baseline model trains on the main task and predicts the target task. The multi-task model uses one or both auxiliary tasks in addition to the main task. The adversarial model uses the adversarial task in addition to the main task. All models use the online disc. dev set for model selection.

4.2 ONLINE DISCUSSION ANNOTATIONS

We create a new resource of issue-frame annotated online fora discussions, by annotating a subset of the Argument Extraction Corpus

(Swanson, Ecker, and Walker, 2015) with a subset of the frames in the Policy Frames Codebook. The Argument Extraction Corpus is a collection of argumentative dialogues across topics and platforms.[2] The corpus contains posts on the following topics: *gay marriage*, *gun control*, *death penalty* and *evolution*. A subset of the corpus was annotated with argument quality scores by Swanson, Ecker, and Walker, (2015), which we exploit in our multi-task setup (see §4.3).

We collect new issue frame annotations for each argument in the argument-quality annotated data.[3] We refer to this new issue-frame annotated corpus as *online discussion corpus* henceforth. Each argument can have one or multiple frames. Following Naderi and Hirst, (2017), we focus on the five most frequent issue frames: *Economic*, *constitutionality and jurisprudence*, *policy prescription and evaluation*, *law and order/crime and justice*, and *political*. See Table 4.1 for examples and Table 4.2 for the class distribution in the resulting online discussions test set. Phrases which do not match the five categories are labeled as *Other*, but we do not consider this class in our experiments. The annotations were done by a single annotator. A second annotator labeled a subset of 200 instances that we use to compute agreement as macro-averaged F-score, assuming one of the annotations as gold standard. Results are 0.73 and 0.7, respectively. The averaged Cohen's Kappa is 0.71.

4.3 ADDITIONAL DATA

The dataset described in the previous section serves as evaluation set for the online discussions domain. As we do not have labeled training data for this domain, we exploit additional corpora and additional annotations, which are described in the next subsection. Statistics of the filtered datasets as well as preprocessing details are given in Appendix 9.

MEDIA FRAMES CORPUS The Media Frames Corpus (Card et al., 2015) contains US newspaper articles on three topics: *Immigration*, *smoking* and *same-sex marriage*. The articles are annotated with the 15 framing dimensions defined in the Policy Frames Codebook.[4] The annotations are on span-level and can cross sentence boundaries. We convert span annotations to sentence-level annotations as follows: if a span annotated with label l lies within sentence boundaries and covers at least 50% of the tokens in the sentence, we label the sentence

2 The corpus is a combination of dialogues from http://www.createdebate.com/, and Walker et al., (2012a)'s Internet Argument Corpus, which contains dialogues from 4forums.com.

3 Topic cluster *Evolution* was dropped, because it contained too few examples matching our frame categories.

4 We discard all instances that do not correspond to the frame categories in the online discussions data.

with l. We only keep sentence annotations if they are indicated by at least two annotators.

CONGRESSIONAL TWEETS DATASET The congressional tweets dataset (Johnson, Jin, and Goldwasser, 2017) contains tweets authored by 40 members of the US Congress, annotated with the frames of the Policy Frames Codebook. The tweets are related to one or two of the following six issues: *abortion*, *the Affordable Care Act*, *gun rights vs. gun control*, *immigration*, *terrorism*, and *the LGBTQ community*, where each tweet is annotated with one or multiple frames.

ARGUMENT QUALITY ANNOTATIONS The corpus of online discussions contains additional annotations that we exploit in the multi-task setup. Swanson, Ecker, and Walker, (2015) sampled a subset of 5,374 sentences, using various filtering methods to increase likelihood of high quality argument occurrence, and collected annotations for argument quality via crowdsourcing. Annotators were asked to rate argument quality using a continuous slider [0-1]. Seven annotations per sentence were collected. We convert these annotations into binary labels (1 if $\geq$ 0.5, 0, otherwise) and generate an approximately balanced dataset for a binary classification task that is then used as an auxiliary task in the multi-task setup. Balancing is motivated by the observation that balanced datasets tend to be better auxiliary tasks (Bingel and Søgaard, 2017).

4.4 MODELS

The task we are faced with is (multi-label) sequence classification for online discussions. However, we have no labeled training data (and only a small labeled validation set) for the target task in the target domain. Hence, we train our model on a dataset which is labeled with the target labels, but from a different domain. The largest such dataset is the news articles corpus, which we consequently use as main task. Our baseline model is a two-layer LSTM (Hochreiter and Schmidhuber, 1997) trained on only the news articles data. We then apply two strategies to facilitate the transfer of information from source to target domain, multi-task learning and adversarial learning. We briefly describe both setups in the following. An overview over tasks and data used in the different models is shown in Table 4.3.

MULTI-TASK LEARNING To exploit synergies between additional datasets/annotations, we explore a simple multi-task learning with hard parameter sharing strategy, pioneered by (Caruana, 1993), introduced in the context of NLP by (Collobert et al., 2011), and to Recurrent Neural Network (RNN)s by (Søgaard and Goldberg, 2016), which has been shown to be useful for a variety of NLP tasks, e.g. sequence la-

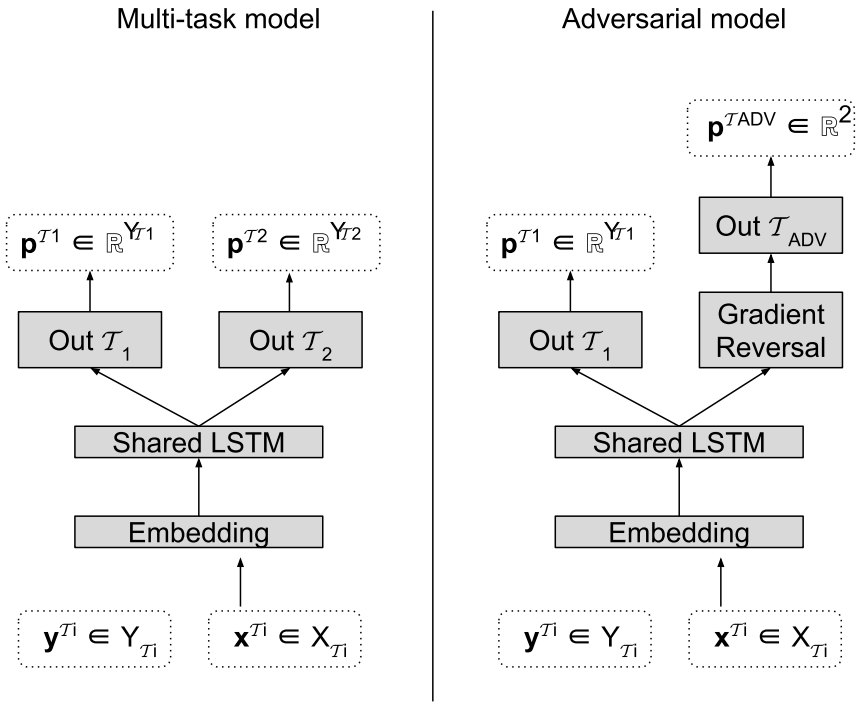

Figure 4.1: Overview over the multi-task model (left) and the adversarial model (right). The baseline LSTM model corresponds to the same architecture with only one task.

belling (Augenstein and Søgaard, 2017; Rei, 2017; Ruder et al., 2019), pairwise sequence classification (Augenstein, Ruder, and Søgaard, 2018) or machine translation (Dong et al., 2015). Here, parameters are shared between hidden layers. Intuitively, it works by training several networks in parallel, tying a subset of the hidden parameters so that updates in one network affect the parameters of the others. By sharing parameters, the networks regularize each other, and the network for one task can benefit from representations induced for the others.

Our multi-task architecture is shown in Figure 4.1. We have N different datasets $\mathcal{T}_1, \cdots, \mathcal{T}_N$. Each dataset $\mathcal{T}_i$ consists of tuples of sequences $x^{\mathcal{T}_i} \in X_{\mathcal{T}_i}$ and labels $y^{\mathcal{T}_i} \in Y_{\mathcal{T}_i}$. A model for task $\mathcal{T}_i$ consists of an input layer, an LSTM layer (that is shared with all other tasks) and a feed forward layer with a softmax activation as output layer. The input layer embeds a sequence $x^{\mathcal{T}_i}$ using pretrained word embeddings. The LSTM layer recurrently processes the embedded sequence and outputs the final hidden state h. The output layer outputs a vector of probabilities $p^{\mathcal{T}_i} \in \mathbb{R}^{Y_{\mathcal{T}_i}}$, based on which the loss $\mathcal{L}_i$ is computed as the categorical cross-entropy between prediction $p^{\mathcal{T}_i}$ and true label $y^{\mathcal{T}_i}$. In each iteration, we sample a data batch for one of the tasks and update the model parameters using stochastic gradient descent. If we sample a batch from the main task or an auxiliary task is decided by a weighted coin flip.

Nr.	Gold	Adv	MTL	LSTM	Sentence
(1)	5	5	5	7	But, star gazer, we had guns then when the Constitution was written and enshrined in the BOR and now incorporated into th 14th Civil Rights Amendment.
(2)	6	6	5	1	Gun control is about preventing such security risks.
(3)	7	7	5	1	First, you warn me of the dangers of using violent means to stop a crime.
(4)	5	6	6	6	So I don't see restrictions on handguns in D.C. as being a clear violation of the Second Amendment.

Table 4.4: Examples for model predictions on the online discussion dev set. The first column shows the gold label and the following columns the prediction made by the adversarial model (Adv), the Multi-Task model (MTL) and the LSTM baseline (LSTM).

ADVERSARIAL LEARNING Ganin and Lempitsky, (2015) proposed adversarial learning for domain adaptation that can exploit unlabeled data from the target domain. The idea is to learn a classifier that is as good as possible at assigning the target labels (learned on the source domain), but as poor as possible in discriminating between instances of the source domain and the target domain. With this strategy, the classifier learns representations that contain information about the target class but abstract away from domain-specific features. During training, the model alternates between 1) predicting the target labels and 2) predicting a binary label discriminating between source and target instances. In this second step, the gradient that is backpropagated is flipped by a Gradient-Reversal layer.[5] Consequently, the model parameters are updated such that the classifier becomes worse at solving the task. The architecture is shown in the right part of Figure 4.1. In our implementation, the model samples batches from the adversarial task or the main task based on a weighted coinflip.

4.5 EXPERIMENTS

We compare the multi-task learning and the adversarial setup with two baseline models: (a) a Random Forest classifier using tf-idf weighted bag-of-words-representations, and (b) the LSTM baseline model. For the multi-task model, we use both the Twitter dataset and the argument quality dataset as auxiliary tasks. For all models, we report results on the test set using the optimal hyper-parameters that we found averaged over 3 runs on the validation set. For the neural models, we use 100-dimensional Global Vectors for Word Representation (GloVe)

5 In the forward pass, this layer multiplies its input with the identity matrix.

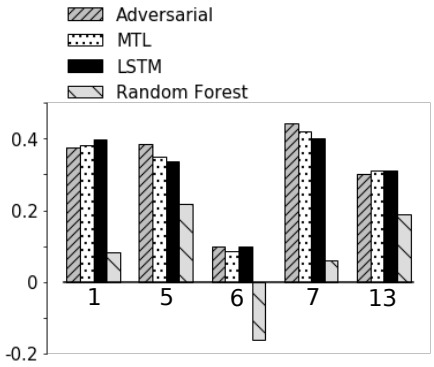

Figure 4.2: Improvement in F-score over the random baseline by class. The absolute F-scores for the best performing system for classes 1, 5, 6, 7, and 13, are 0.529, 0.625, 0.298, 0.655, and 0.499, respectively.

embeddings (Pennington, Socher, and Manning, 2014), pre-trained on Wikipedia and Gigaword.[6] Details about hyper-parameter tuning and optimal settings can be found in Appendix 9.

Model	P_{ma}	R_{ma}	F_{ma}	F_{mi}
Random Baseline	0.196	0.198	0.189	0.196
Random Forest Baseline	0.496	0.335	0.267	0.279
LSTM Baseline	0.512	0.510	0.503	0.521
Multi-Task	0.526	0.525	0.505	0.534
Adversarial	**0.533**	**0.534**	**0.515**	**0.548**

Table 4.5: Macro- (*ma*) and micro-averaged (*mi*) scores for the online discussion test data averaged over 3 runs. The multi-task model uses the Twitter and argument quality datasets as auxiliary tasks. The micro-average F of a baseline that predicts the majority class is 0.307.

RESULTS The results in Table 4.5 show that both the multi-task and the adversarial model improve over the baselines. The multi-task model achieves minor improvements over the LSTM baseline, with a bigger improvement in the micro-averaged score, indicating bigger improvements with frequent labels. The adversarial model performs best, with an error reduction in micro-averaged F over the LSTM baseline of 5.6%.

Figure 4.2 shows the system performances for each class. Each bar indicates the difference between the F-score of the respective system and the random baseline. The adversarial model achieves the biggest improvements over the baseline for classes 5 and 7, which are the

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

two most frequent classes in the test set (cf. Table 9.1 in Appendix 9). For classes 1 and 13, the adversarial model is outperformed by the LSTM. Furthermore, we see that the hardest frame to predict is the *Policy prescription and evaluation frame* (6), where the models achieve the lowest improvement over the baseline and the lowest absolute F-score. This might be because utterances with this frame tend to address specific policies that vary according to topic and domain of the data, and are thus hard to generalize from source to target domain.

ANALYSIS Table 4.4 contains examples of model predictions on the dialogue dev set. In Example (1), the adversarial and the multi-task model correctly predict a *Constitutionality* frame, while the LSTM model incorrectly predicts a *Crime and punishment* frame. In Examples (2) and (3), only the adversarial model predicts the correct frames. In both cases, the LSTM model incorrectly predicts an *Economic* frame, possibly because it is misled by picking up on a different sense of the terms *means* and *risks*. In Example (4), all models make an incorrect prediction. We speculate this might be because the models pick up on the phrase *restrictions on handguns* and interpret it as referring to a policy, whereas to correctly label the sentence they would have to pick up on the *violation of the Second Amendment*, indicating a *Constitutionality* frame.

4.6 CONCLUSION

This work introduced a new benchmark of political discussions from online fora, annotated with issue frames following the Policy Frames Cookbook. Online fora are influential platforms that can have impact on public opinion, but the language used in such fora is very different from newswire and other social media. We showed, however, how multi-task and adversarial learning can facilitate transfer learning from such domains, leveraging previously annotated resources to improve predictions on informal, multi-party discussions. Our best model obtained a micro-averaged F1-score of 0.548 on our new benchmark.

ACKNOWLEDGEMENTS

We acknowledge the resources provided by CSC in Helsinki through NeIC-NLPL (www.nlpl.eu), and the support of the Carlsberg Foundation and the NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

Part III

CROSS-LINGUAL WORD REPRESENTATIONS

5

LIMITATIONS OF CROSS-LINGUAL LEARNING FROM IMAGE SEARCH

ABSTRACT

Cross-lingual representation learning is an important step in making NLP scale to all the world's languages. Previous work on bilingual lexicon induction suggests that it is possible to learn cross-lingual representations of words based on similarities between images associated with these words. However, that work focused (almost exclusively) on the translation of nouns only. Here, we investigate whether the meaning of other POS, in particular adjectives and verbs, can be learned in the same way. Our experiments across five language pairs indicate that previous work does not scale to the problem of learning cross-lingual representations beyond simple nouns.

5.1 INTRODUCTION

Typically, cross-lingual word representations are learned from word alignments, sentence alignments, from aligned, comparable documents Levy, Goldberg, and Søgaard, 2017, or from monolingual corpora using seed dictionaries (Ammar et al., 2016).[1] However, for many languages such resources are not available.

(Bergsma and Van Durme, 2011) introduced an alternative idea, namely to learn bilingual representations from image data collected via web image search. The idea behind their approach is to represent words in a visual space and find valid translations between words based on similarities between their visual representations. Representations of words in the visual space are built by representing a word by a set of images that are associated with that word, i.e., the word is a semantic tag for the images in the set.

(Kiela, Vulic, and Clark, 2015) improve performance for the same task using a feature representation extracted from convolutional networks. However, both works only consider nouns, leaving open the question of whether learning cross-lingual representations for other POS from images is possible.[2]

1 Recent work by (Conneau et al., 2018) introduces unsupervised bilingual lexicon induction from monolingual corpora, however, it was shown that this approach has important limitations (Søgaard, Ruder, and Vulić, 2018).

2 (Kiela, Verő, and Clark, 2016) induce English-Italian word translations from image data for the Simlex-999 dataset which contains adjectives and verbs, but they do not evaluate the performance for these POS compared to nouns.

In order to evaluate whether this work scales to verbs and adjectives, we compile wordlists containing these POS in several languages. We collect image sets for each image word and represent all words in a visual space. Then, we rank translations computing similarities between image sets and evaluate performance on this task.

Another field of research that exploits image data for NLP applications is the induction of multi-modal embeddings, i.e. semantic representations that are learned from textual and visual information jointly (Hill and Korhonen, 2014; Kiela and Bottou, 2014; Kiela, Verő, and Clark, 2016; Kiela et al., 2014; Lazaridou, Pham, and Baroni, 2015; Silberer, Ferrari, and Lapata, 2017; Vulić et al., 2016). The work presented in our paper differs from these approaches, in that we do not use image data to improve semantic representations, but use images as a resource to learn cross-lingual representations. Even though lexicon induction from text resources might be more promising in terms of performance, we think that lexicon induction from visual data is worth exploring as it might give insights in the way that language is grounded in visual context.

5.1.1 CONTRIBUTIONS

We evaluate the approaches by (Bergsma and Van Durme, 2011) and (Kiela, Vulic, and Clark, 2015) on an extended data set, which apart from nouns includes both adjectives and verbs. Our results suggest that none of the approaches involving image data are directly applicable to learning cross-lingual representations for adjectives and verbs.

5.2 DATA

WORDLISTS We combined 3 data sets of English words to compile the wordlists for our experiments: the original wordlist used by (Kiela, Vulic, and Clark, 2015), the Simlex-999 data set of English word pairs (Hill, Reichart, and Korhonen, 2014) and the A dataset for multimodal distributional semantics (MEN) data set (Bruni, Tran, and Baroni, 2014). Whereas the first wordlist contains only nouns, the latter two datasets contain words of three POS classes (nouns, adjectives and verbs). We collect all distinct words and translate the final wordlist into 5 languages (German, French, Russian, Italian, Spanish) using the Google translation API[3], choosing the most frequent translation with the respective POS tag. Table 5.1 shows the POS distribution in the datasets.

3 https://translate.google.com/

(a) Images associated with the English noun *cow* (left) and the German translation *Kuh* (right).

(b) Images associated with the English verb *discuss* (left) and the German translation *diskutieren* (right).

(c) Images associated with the English adjective *sad* (left) and the German translation *traurig* (right).

Figure 5.1: Examples for images associated with equivalent words in two languages (English and German).

IMAGE DATA SETS We use the Google Custom Search API[4] to represent each word in a wordlist by a set of images. We collect the first 50 jpeg images returned by the search engine when querying the words specifying the target language.[5] This way, we compile image data sets for 6 languages.[6] Figure 5.1 shows examples for images associated with a word in two languages.

4 https://developers.google.com/custom-search/

5 Even though we get the search results for the first 50 images, some of them cannot be downloaded. On average, we collect 42 images for each image word.

6 The wordlists and image datasets are available at https://github.com/coastalcph/cldi_from_image_search/

	MEN	Simlex	Bergsma	Combined
N	656	751	500	1406
V	38	170	0	206
A	57	107	0	159

Table 5.1: Distribution of POS tags in the datasets used to compile the final wordlist.

5.3 APPROACH

The assumption underlying the approach is that semantically similar words in two languages are associated with similar images. Hence, in order to find the translation of a word, e.g. from English to German, we compare the images representing the English word with all the images representing German words, and pick as translation the German word represented by the most similar images. To compute similarities between images, we compute cosine similarities between their feature representations.

5.3.1 CONVOLUTIONAL NEURAL NETWORK FEATURE REPRESENTATIONS

Following (Kiela, Vulic, and Clark, 2015), we compute convolutional neural network (CNN) feature representations using a model pre-trained on the ImageNet classification task (Russakovsky et al., 2015). For each image, we extract the pre-softmax layer representation of the CNN. Instead of an AlexNet (Krizhevsky, Sutskever, and Hinton, 2012) as used by (Kiela, Vulic, and Clark, 2015), we use the Keras implementation of the Very deep convolutional network for large-scale image recognition (VGG19) model as described in (Simonyan and Zisserman, 2014), which was shown to achieve similar performance for word representation tasks by (Kiela, Verő, and Clark, 2016). Using this model, we represent each image by a 4069-dimensional feature vector.

SIMILARITIES BETWEEN INDIVIDUAL IMAGES (Bergsma and Van Durme, 2011) determine similarities between image sets based on similarities between all individual images. For each image in image set 1, the maximum similarity score for any image in image set 2 is computed. These maximum similarity scores are then either averaged (AVGMAX) or their maximum is taken (MAXMAX).

SIMILARITIES BETWEEN AGGREGATED REPRESENTATIONS In addition to the above described methods, (Kiela, Vulic, and Clark, 2015) generate an aggregated representation for each image set and then

compute the similarity between image sets by computing the similarity between the aggregated representations. Aggregated representations for image sets are generated by either taking the component-wise average (CNN-MEAN) or the component-wise maximum (CNN-MAX) of all images in the set.

K-NEAREST NEIGHBOR For each image in an image set in language 1, we compute its nearest neighbor across all image sets in language 2. Then, we find the image set in language 2 that contains the highest number of nearest neighbors. The image word is translated into the image word that is associated with that image 2 set. Ties between image sets containing an equivalent number of nearest neighbors are broken by computing the average distance between all members. We refer to the method as K NEAREST NEIGHBOR (KNN). Whereas the other approaches described above provide a ranking of translations, this method determines only the one translation that is associated with the most similar image set.

CLUSTERING IMAGE SETS As we expect the retrieved image sets for a word to contain images associated with different senses of the word, we first cluster images into k clusters. This way, we hope to group images representing different word senses. Then, we apply the KNN method as described above. We refer to this method as KNN-C.

5.3.2 EVALUATION METRICS

Ranking performance is evaluated by computing the Mean Reciprocal Rank (MRR) as $MRR = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{rank(w_s, w_t)}$ M is the number of words to be translated and $rank(w_s, w_t)$ is the position the correct translation w_t for source word w_s is ranked on.

In addition to MRR, we also evaluate the cross-lingual representations by means of P@k.

	ALL			NN			VB			ADJ		
	MRR	P@1	P@10	MRR	P@1	P@10	MRR	P@1	P@10	MRR	P@1	P@10
AVGMAX	0.53	0.49	0.60	0.60	0.56	0.67	0.20	0.15	0.30	0.28	0.22	0.37
MAXMAX	0.44	0.38	0.54	0.49	0.43	0.61	0.19	0.15	0.24	0.23	0.18	0.31
CNNMEAN	0.49	0.44	0.57	0.56	0.52	0.64	0.15	0.10	0.26	0.24	0.20	0.32
CNNMAX	0.47	0.43	0.55	0.55	0.50	0.63	0.15	0.10	0.24	0.19	0.15	0.27
KNN	–	0.42	–	–	0.50	–	–	0.06	–	–	0.13	–
KNN-C	–	0.47	–	–	0.56	–	–	0.10	–	–	0.16	–

Table 5.2: Results for translation ranking with images represented by CNN features averaged over 5 language pairs. KNN and KNN-C do not produce a ranking, hence we only provide P@1 values. For both KNN models, $k = 3$.

5.4 EXPERIMENTS AND RESULTS

We run experiments for 5 language pairs English–German, English–Spanish, English–French, English–Russian and English–Italian. We evaluate the representations computed from image data and compare the different methods for similarity computation described in 5.3. For each English word, we rank all the words in the corresponding target languages based on similarities between image sets and evaluate the models' ability to identify correct translations, i.e. to rank the correct translation on a position near the top. We compare 4 settings that differ in the set of English words that are translated. In the setting ALL, all English words in the wordlist are translated. NN, VB and ADJ refer to the settings where only nouns, verbs and adjectives are translated.

5.4.1 RESULTS

COMPARISON OF SIMILARITY COMPUTATION METHODS FOR VISUAL REPRESENTATIONS Table 5.2 displays results averaged over all language pairs.[7] First, comparing the different methods to compute similarities between image sets, AVGMAX outperforms the other methods in almost all cases. Most importantly, we witness a very significant drop in performance when moving from nouns to verbs and adjectives. For verbs, we rarely pick the right translation based on the image-based word representations. This behavior applies across all methods for similarity computation. Further, we see small improvements if we cluster the image sets prior to applying the KNN method, which might indicate that the clustering helps in finding translations for polysemous words.

5.4.2 ANALYSIS

If we try to learn translations from images, integrating verbs and adjectives into the dataset worsens results compared to a dataset that contains only nouns. One possible explanation is that images associated with verbs and adjectives are less suited to represent the meaning of a concept than images associated with nouns.

(Kiela, Vulic, and Clark, 2015) suppose that lexicon induction via image similarity performs worse for datasets containing words that are more abstract. In order to approximate the degree of abstractness of a concept, they compute the *image dispersion d* for a word *w* as

7 We also evaluate our visual representations on the set of 500 nouns used by Kiela, Vulic, and Clark, (2015), which results in P@1=0.6 and MRR=0.63 averaged over 5 language pairs for the AVGMAX method.

the average cosine distance between all image pairs in the image set $\{i_j, \ldots, i_n\}$ associated with word w according to

$$d(w) = \frac{2}{n(n-1)} \sum_{k<j\leq n} 1 - \frac{i_j \cdot i_k}{|i_j||i_k|}$$

In their analysis, (Kiela, Vulic, and Clark, 2015) find that their model performs worse on datasets with a higher average image dispersion. (Kiela et al., 2014) introduce a dispersion-based filtering approach for learning multi-modal representations of nouns. They show that the quality of their representations with respect to a monolingual word-similarity prediction task improves, if they include visual information only in cases where the dispersion of the visual data is low.

Computing the average image dispersion for our data across languages shows that image sets associated with verbs and adjectives have a higher average image dispersion than image sets associated with nouns (nouns: $d = 0.60$, verbs: $d = 0.68$, adjectives: $d = 0.66$).

Table 5.3 shows the image words associated with the image sets that have the highest and lowest dispersion values in the English image data. For nouns and adjectives, we observe that the words with lowest dispersion values express concrete concepts, whereas the words with highest dispersion values express more abstract concepts that can be displayed in many variants. Manually inspecting the dataset, we find e.g. that the images associated with the noun *animal* display many different animals, such as birds, dogs, etc, whereas the images for *mug* all show a prototypical mug.

	Lowest dispersion		Highest dispersion	
	Word	d	Word	d
NN	mug	0.31	animal	0.78
	oscilloscope	0.32	companion	0.78
	padlock	0.33	mammal	0.78
VB	vanish	0.43	differ	0.76
	shed	0.43	hang	0.76
	divide	0.47	arrange	0.75
ADJ	yellow	0.39	huge	0.79
	white	0.40	large	0.79
	fragile	0.43	big	0.78

Table 5.3: English image words associated with the image sets with highest and lowest dispersion scores d.

Besides the dispersion values, we also analyze the number of word senses per POS using WordNet[8]. We find that the verbs in our dataset have a higher average number of word senses ($n = 9.18$) than the adjectives ($n = 6.88$) and the nouns ($n = 5.08$). That we get worst results for the words with highest number of different word senses is in agreement with (Gerz et al., 2016), who find that in a monolingual word similarity prediction task, models perform worse for verbs with more different senses than for less polysemous verbs.

5.5 CONCLUSION

We showed that existing work on learning cross-lingual word representations from images obtained via web image search does not scale to other POS than nouns. It is possible that training convolutional networks on different resources than ImageNet data will provide better features representing verbs and adjectives. Finally, it would be interesting to extend the approach to multi-modal input, combining images and texts, e.g. from comparable corpora with images such as Wikipedia.

8 https://wordnet.princeton.edu/

6

WHY IS UNSUPERVISED ALIGNMENT OF ENGLISH EMBEDDINGS FROM DIFFERENT ALGORITHMS SO HARD?

ABSTRACT

This paper presents a challenge to the community: Generative adversarial networks (GANs) can perfectly align independent English word embeddings induced using *the same* algorithm, based on distributional information alone; but fails to do so, for two different embeddings algorithms. *Why is that?* We believe understanding why, is key to understand *both* modern word embedding algorithms *and* the limitations and instability dynamics of GANs. This paper shows that (a) in all these cases, where alignment fails, there exists a linear transform between the two embeddings (so algorithm biases do not lead to non-linear differences), and (b) similar effects can not easily be obtained by varying hyper-parameters. One plausible suggestion based on our initial experiments is that the differences in the inductive biases of the embedding algorithms lead to an optimization landscape that is riddled with local optima, leading to a very small basin of convergence, but we present this more as a challenge paper than a technical contribution.

6.1 INTRODUCTION

This paper brings together two fascinating research topics in NLP, namely *understanding the properties of word embeddings* (Mikolov et al., 2013; Mimno and Thompson, 2017; Mitchell and Steedman, 2015) and *unsupervised bilingual dictionary induction* (Conneau et al., 2018; Søgaard, Ruder, and Vulić, 2018; Zhang et al., 2017b). In an effort to better understand when unsupervised bilingual dictionary induction is possible, we factored out linguistic differences between languages, and studied English-English alignability (by learning to align English embeddings trained on different samples of the English Wikipedia), when we came across a puzzling phenomena: *English-English can be aligned with almost 100% precision, if you use the same embedding algorithms for the two samples, but not at all (0% precision), if you use different embedding algorithms*. This results suggest that the properties of word embeddings induced by different algorithms challenge unsupervised bilingual dictionary algorithms. Understanding why will enable us to develop more stable adversarial learning algorithms and give us a better understanding of how embedding algorithms differ.

CONTRIBUTIONS We are, to the best of our knowledge, the first to study unsupervised alignability of pairs of English word embeddings. We show that unsupervised alignment – specifically the MUSE system (Conneau et al., 2018) – fails when the algorithms used to induce the two embeddings differ, and that this is *not* because there is no linear transformation between the two spaces. We further show that poor initialization, as a result of MUSE initially applying an identity transform to two word embeddings far apart in space, is not the sole reason the discriminator suffers from local optima. Finally, we present an experiment showing what the minimal corpus size is for unsupervised alignment to succeed, in the absence of linguistic differences.

6.2 ALIGNING EMBEDDINGS

6.2.1 UNSUPERVISED ALIGNMENT USING GENERATIVE ADVERSARIAL NETWORKS

MUSE (Conneau et al., 2018) uses a vanilla GAN with a linear generator to learn alignments between embedding spaces without supervision. In a two-player game, a discriminator D aims to tell the two language spaces apart, while a generator G aims to map the source language into the target language space, fooling the discriminator. While MUSE achieves impressive results at times, MUSE is highly unstable, e.g., with different initializations precision scores vary between 0% and 45% for English-Greek (Søgaard, Ruder, and Vulić, 2018).

The parameters of a GAN with a linear generator are (Ω, w). They are obtained by solving the following min-max problem:

$$\min_{\Omega} \max_{w} \mathrm{E}[\log(D_w(X)) + \log(1 - D_w(g_\Omega(Z)))] \tag{6.1}$$

which reduces to

$$\min_{\Omega} JS(P_X \mid P_\Omega) \tag{6.2}$$

Ω is initialized as the identity matrix I.

If G wins the game against an ideal discriminator on a very large number of samples, then F (the source vector space) and ΩE (with E being the target vector space) can be shown to be close in Jensen-Shannon divergence, and thus the model has learned the true distribution. This result, referring to the distributions of the data, p_{data}, and the distribution, p_g, G is sampling from, is from Goodfellow et al., (2014): *If G and D have enough capacity, and at each step of training, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion*

$$E_{\mathbf{x}\sim p_{data}}[\log D^*_G(\mathbf{x})] + E_{\mathbf{x}\sim p_g}[\log(1 - D^*_G(\mathbf{x}))]$$

then p_g converges to p_{data}.

This result relies on a number of assumptions that do not hold in practice. Our generator, which learns a linear transform Ω, has very limited capacity, for example, and we are updating Ω rather than p_g. In practice, therefore, during training, we alternate between k steps of optimizing the discriminator and one step of optimizing the generator. If the GAN-based alignment is not successful, this can thus be a result of two things: Either that G does not have enough capacity, or that D is stuck in a local optimum. Our results in Section 6.3 show that the inability to align English-English in the case of different word embedding algorithms is *not* a result of limited capacity, but a result of the GAN being trapped in one of the many local optima of the loss function.

6.2.2 SUPERVISED ALIGNMENT USING PROCRUSTES ANALYSIS

Procrustes Analysis (Schönemann, 1966) has been commonly used for supervised alignment of word embeddings (Artetxe, Labaka, and Agirre, 2018b; Smith et al., 2017). Here, the optimal alignment between two embedding spaces is computed using singular value decomposition of the aligned embeddings in a seed dictionary. Conneau et al., (2018) use Procrustes Analysis to refine an initial seed dictionary learned by the generative adversarial network without supervision. In our supervised experiments, we use 5000 seed words as supervision for learning the alignment between embeddings.

6.2.3 GEOMETRY OF EMBEDDINGS

Below we summarize some previous findings about the geometry of monolingual embeddings (Mimno and Thompson, 2017), and add some new observations. We discuss five embedding algorithms: Singular Value Decomposition (SVD) on positive Pointwise Mutual Information (PMI) matrices (Hyperwords-SVD) (Levy, Goldberg, and Dagan, 2015), skip-gram negative sampling applied to co-occurrence matrices (Hyperwords-Skipgram Negative Sampling (SGNS)) (Levy, Goldberg, and Dagan, 2015), Continuous Bag-of-Words (CBOW) (Mikolov et al., 2013a), GloVe (Pennington, Socher, and Manning, 2014), and FastText (Bojanowski et al., 2017). To analyze the geometry of our monolingual embeddings in space, we report average inner product to mean vector; see (Mimno and Thompson, 2017) for details.

HYPERWORDS-SVD have a small average inner product (0.0032), suggesting they are well-dispersed through space; like Hyperwords-SGNS and standard SGNS (Mimno and Thompson, 2017), they do not exhibit a clear word frequency bias. **Hyperwords-SGNS** vectors also have a small average inner product (0.0002), in contrast with

	Hyperwords-SGNS	Hyperwords-SVD	CBOW	GloVe	FastText
			UNSUPERVISED		
Hyperwords-SGNS	**0.997**				
Hyperwords-SVD	0.000	**0.992**			
CBOW	0.000	0.000	**0.997**		
GloVe	0.000	0.000	0.000	**0.997**	
FastText	0.000	0.000	0.000	0.000	**0.997**
			SUPERVISED		
Hyperwords-SVD	0.967				
CBOW	0.990	0.989			
GloVe	0.985	0.992	0.999		
FastText	0.994	0.994	0.999	0.997	

Table 6.1: Precision at 1 (P@1) for unsupervised GAN alignment with Procrustes refinement (top) and supervised Procrustes analysis for the cases in which unsupervised alignment fails (bottom). Results clearly show that GANs can align two independent embeddings induced by the same algorithm; but not embeddings aligned by different ones. Supervised Procrustes analysis, on the other hand, perfectly aligns the embeddings in both cases.

standard SGNS vectors, which are narrowly clustered in a single orthant (Mimno and Thompson, 2017). In line with standard SGNS vectors, the frequency of words has relatively little effect on their inner product, with the exception of the rare words, which have slightly less positive inner products. **CBOW** vectors have a relatively large average inner product (4.2985). The vectors trained by **GloVe** show a clear relationship with word frequency, with low-frequency words opposing the frequency-balanced mean vector. The embeddings are well-dispersed, with an average inner product of 0.0002. Finally, **FastText** vectors have a large, positive inner product with the mean (0.2988), indicating that they are not evenly dispersed through the space, but pointing in roughly the same direction. The FastText vectors exhibit a frequency bias, much like what has been previously observed with GloVe vectors. The differences are the results of the inductive biases of the different embedding algorithms.

6.3 EXPERIMENTS

This section presents our data, the hyper-parameters of our embeddings, our experimental protocols, and our results.

6.3.1 DATA

In the following experiments we learn word embeddings on samples of a publicly available Wikipedia dump from March 2018.[1] The data is preprocessed using a publicly available preprocessing script[2], extracting text, removing non-alphanumeric characters, converting digits to text, and lowercasing the text.

6.3.2 HYPER-PARAMETERS

We train 300-dimensional word embeddings using the algorithms' recommended hyperparameter settings, listed in the following:[3] For **Hyperwords-SGNS**, the window size is set to 2 and the subsampling of frequent words and smoothing of the context distribution are disabled. The minimal word count for being in the vocabulary is 100. The same applies for **Hyperwords-SVD**, and the exponent for weighting the eigenvalue matrix is 0.5. For **CBOW**, the window size is set to 8, the number of negative samples is 25, and the subsampling threshold for frequent words is 1e-4. For **GloVe**, the window size is set to 15 and the cutoff parameter x_{max} to 10. Finally, for **FastText**, the window size is 5, the number of negatives samples is 5 and the sampling threshold is 0.0001.

6.3.3 MAIN EXPERIMENTS

We train word embeddings using the different embedding algorithms listed in Section 6.3.2 on two non-overlapping 10% samples of the English Wikipedia dump (the samples contain 463,576 and 528,556 distinct words, with an overlap in vocabulary of 351,858 words). We learn unsupervised and supervised alignments for embeddings (as described in Section 6.2) trained by different algorithms on the same datasplits, and for embeddings trained by the same algorithm on the two different datasplits. For the unsupervised alignments, we use the default parameters of the MUSE system for the adversarial training, i.e. a discriminator with 2 fully connected layers of 2048 units trained over 5 epochs, 1,000,000 iterations per epoch with 5 discriminator steps per iteration and a batch size of 32.

We evaluate the alignments in terms of Precision@1 in the word translation retrieval task for the 1500 test words used by Conneau et al., (2018). The results are shown in Table 6.1[4]. Our main observations are: (a) MUSE learns perfect alignments for embeddings learned by the same

1 https://dumps.wikimedia.org/enwiki/

2 http://mattmahoney.net/dc/textdata.html

3 We also ran experiments with one of the embedding algorithms (FastText) to check if our results were robust across hyper-parameter settings

4 We report Precision at 1 scores but find that the pattern is the same for Precision at 10, with perfect alignments for embeddings from the same algorithm and 0 scores

algorithm on different data splits. (b) MUSE cannot learn alignments for embeddings learned by different algorithms on the same data splits, even if there exists a linear transformation aligning both sets of embeddings (the supervised algorithm learns perfect alignments). We also verify that MUSE cannot learn to align embeddings from different algorithms *even when induced from the same sample*. As already mentioned, we also ran experiments to check that the failure of MUSE to learn good alignments was not a result of the differences in hyper-parameter settings. Section 6.3.4 presents additional experiments with normalization, for control; Section 6.3.5 addresses how much data is needed to align independently induced embeddings from the same algorithm. Section 6.4 discusses potential answers to why MUSE fails when embeddings are induced using different algorithms.

6.3.4 EXPERIMENTS WITH NORMALIZATION

The embeddings in the main experiments differ in several ways; see Section 6.2. One possible explanation for the inability of MUSE to align embeddings from different algorithms could be that the two embeddings are so far apart in space that the discriminator learns to discriminate between them too quickly. Recall that Ω is initialized as the identity matrix I, which means that the generator initially presents the discriminator with the source embedding as is. This is an effect that has often been observed with GANs (Arjovsky and Bottou, 2017); could this also be the explanation for our results? At a first glance, this seems a possible explanation. The inner products with the mean differ significantly for the five embedding algorithms (see Section 6.2). The only embeddings that have roughly the same directionality are Hyperwords and GloVe, and their centroids are very far apart in cosine space. The cosine similarity of the centroids of the two versions of Hyperwords is -0.006, and the cosine similarity for Hyperwords-SVD and GloVe is 0.019. However, poor initialization as a result of applying the identity transform to very distant word embeddings is not the explanation for the poor performance of MUSE in this set-up: Both sets of Hyperwords embeddings were normalized, but alignment still failed. To verify this holds in general, i.e., that results are not affected by normalization in general, we also ran experiments with the remaining 14 embedding pairs, normalizing and/or centering both embeddings. Results stayed the same: Precision at 1 scores of 0.

6.3.5 LEARNING CURVE

MUSE perfectly aligns independently induced word embeddings induced by the same algorithm. For FastText, it correctly aligns 99.7% of

for alignments between embeddings from different algorithms in the unsupervised experiments.

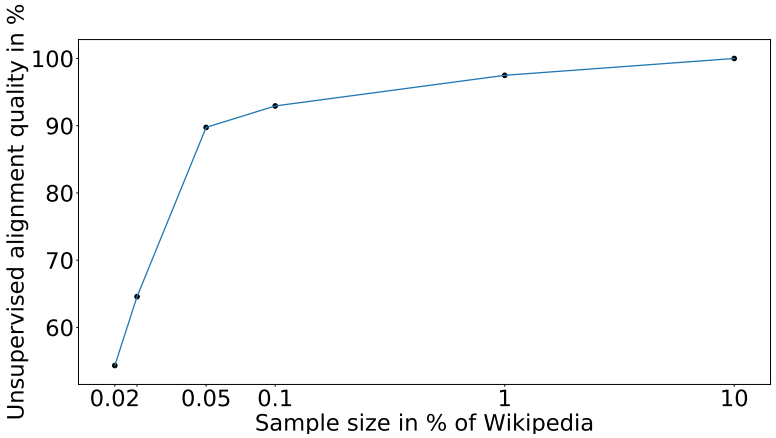

Figure 6.1: Unsupervised alignment quality for FastText embeddings trained on samples of different sizes, evaluated on 878 words covered by all of the embeddings. The x-axis is log-scaled.

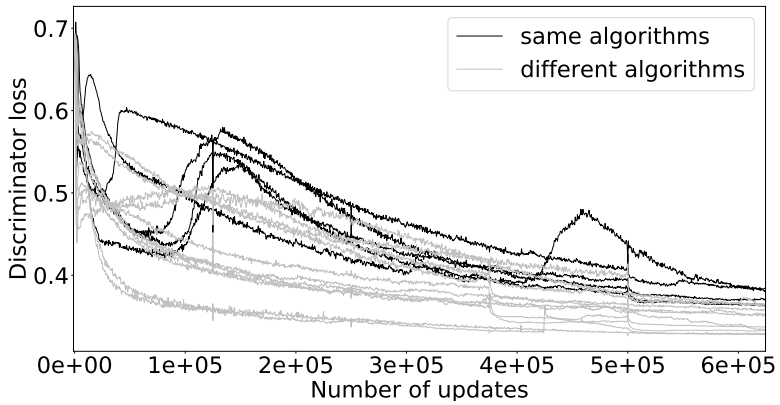

Figure 6.2: Discriminator losses using the same algorithm for source and target (black curves) or using different algorithms (grey curves).

all words in the evaluation lexicon with itself. Our samples are 10% of a publicly available Wikipedia dump, amounting to more than 400 million tokens per sample. English-English alignment is an interesting control experiment for unsupervised bilingual dictionary induction, abstracting away from linguistic differences, and we ran a series of experiments to see how small samples MUSE can align in the absence of linguistic differences. The learning curve is presented in Figure 6.1.

6.4 DISCUSSION

We have shown that the fact that MUSE cannot align two embedding spaces for English induced by different algorithms (even if using the same corpus), is *not* a result of there not being a linear transformation, and not a result of (lack of) normalization or trivial differences in

model hyper-parameters. The only explanation left seems to be that the inductive biases of the different algorithms lead to a loss landscape so riddled with local optima that MUSE cannot possible escape them.

To support this hypothesis, compare the loss curves for the MUSE runs aligning embeddings induced with the *same* algorithms (black curves) to the runs aligning embeddings induced with different algorithms, in Figure 6.2. When the embeddings are induced by the same algorithm, we clearly see the contours of a min-max game, suggesting that the generator and discriminator challenge each other, both contributing to a good alignment. When the embeddings are induced by different algorithms, however, the discriminator quickly drops, with the generator unable to push the discriminator out of a local optimum. *Understanding when biases induce highly non-convex landscapes, and how to make adversarial training less sensitive to such scenarios, remains an open problem, which we think will be key to the success of unsupervised machine translation and related tasks.*

7

COMPARING UNSUPERVISED WORD TRANSLATION METHODS STEP BY STEP

ABSTRACT

Cross-lingual word vector space alignment is the task of mapping the vocabularies of two languages into a shared semantic space, which can be used for dictionary induction, unsupervised machine translation, and transfer learning. In the unsupervised regime, an initial seed dictionary is learned in the absence of any known correspondences between words, through **distribution matching**, and the seed dictionary is then used to supervise the induction of the final alignment in what is typically referred to as a (possibly iterative) **refinement** step. We focus on the first step and compare distribution matching techniques in the context of language pairs for which mixed training stability and evaluation scores have been reported. We show that, surprisingly, when looking at this initial step in isolation, vanilla GANs are superior to more recent methods, both in terms of precision and robustness. The improvements reported by more recent methods thus stem from the refinement techniques, and we show that we can obtain state-of-the-art performance combining vanilla GANs with such refinement techniques.

7.1 INTRODUCTION

A word vector space – sometimes referred to as a *word embedding* – associates similar words in a vocabulary with similar vectors. Learning a projection of one word vector space into another, such that similar words – across the two word embeddings – are associated with similar vectors, is useful in many contexts, with the most prominent example being the alignment of vocabularies of different languages, i.e., word translation. This is a key step in machine translation of low-resource languages (Lample, Denoyer, and Ranzato, 2018).

Projections between word vector spaces have typically been learned from seed dictionaries. In seminal papers (Faruqui and Dyer, 2014; Gouws and Søgaard, 2015; Mikolov, Le, and Sutskever, 2013), these seeds would comprise thousands of words, but Vulić and Korhonen, (2016) showed that we can learn reliable projections from as little as 50 words. Smith et al., (2017) and Hauer, Nicolai, and Kondrak, (2017) subsequently showed that the seed can be replaced with just words that are identical across languages; and Artetxe, Labaka, and Agirre, (2017) showed that numerals can also do the job, in some

cases; both proposals removing the need for an actual dictionary. Even more recently, entirely unsupervised approaches to projecting word vector spaces onto each other have been proposed, which induce seed dictionaries in the absence of any known correspondences between words, using distribution matching techniques. These seed dictionaries are then used as supervision for alignment algorithms based on, e.g., Procrustes Analysis Schönemann, 1966. These unsupervised systems, in other words, typically combine two steps: an unsupervised step of distribution matching and a (possibly iterative) (pseudo-)supervised step of refinement, based on a seed dictionary learned in the first step. See Table 7.1 for an overview.

The first Unsupervised Bilingual Dictionary Induction (UBDI) systems (Barone, 2016; Conneau et al., 2018; Zhang et al., 2017b) were based on GANs (Goodfellow et al., 2014). These approaches learn a linear transformation to minimize the divergence between a target distribution (say French word embeddings) and a source distribution (the English word embeddings projected into the French space). GAN-based approaches achieve impressive results for some language pairs (Conneau et al., 2018), but show instabilities for others. In particular, Søgaard, Ruder, and Vulić, (2018) presented results suggesting that GAN-based UBDI is difficult for some language pairs exhibiting very different morphosyntactic properties, as well as when the monolingual corpora are very different. Recently, a range of unsupervised approaches that do not rely on GANs have been proposed (Artetxe, Labaka, and Agirre, 2018a; Grave, Joulin, and Berthet, 2018; Hoshen and Wolf, 2018) in the hope they would provide a more robust alternative. In this paper, we show *none of these are more robust* on the language pairs we consider. Instead we propose a simple technique for making (vanilla) GAN-based UBDI more robust and show that combining this with a recently proposed refinement technique – stochastic dictionary induction (Artetxe, Labaka, and Agirre, 2018a) – leads to state-of-the-art performance in UBDI.

CONTRIBUTIONS We present the first systematic comparison of (a subset of) recently proposed methods for UBDI. These methods are two-step pipelines of unsupervised distribution matching for seed induction and supervised refinement. While the authors typically introduce new approaches to both steps (see Table 7.1), distribution matching and refinement are independent, and in this paper, **we focus on the distribution matching step** - by either omitting refinement or using the same refinement method across different distribution matching, or seed dictionary induction methods. On the language pairs considered here, vanilla GANs are superior to more recently improved distribution matching techniques. Moreover, we show that using an unsupervised model selection method, we can often pick out the best vanilla GAN runs *in the absence of* cross-lingual supervision.

	Initialization and optimization steps		
Authors	**Unsupervised step**	**Supervised step**	**Extras**
Barone, 2016	GAN	None	
Zhang et al., 2017b	Wasserstein GAN	Procrustes	
Conneau et al., 2018	GAN	Procrustes	
Hoshen and Wolf, 2018	ICP	Procrustes	Restarts
Alvarez-Melis and Jaakkola, 2018	Gromov-Wasserstein	Procrustes	
Artetxe, Labaka, and Agirre, 2018a	Gromov-Wasserstein	Stochastic	
Yang et al., 2018	Gromov-Wasserstein	MMD	
Xu et al., 2018	GAN	Sinkhorn	Back-translation
Grave, Joulin, and Berthet, 2018	Gold-Rangarajan	Sinkhorn	

Table 7.1: Approaches to unsupervised alignment of word vector spaces. We break down these approaches in two steps (and extras): (1) **Unsupervised** distribution matching for seed dictionary learning): (W)GANs, ICP, Gromov-Wasserstein initialization, and the convex relaxation proposed in Gold and Rangarajan, 1996. (2) **Supervised** refinement: Procrustes, stochastic dictionary induction, maximum mean discrepancy (MMD), and the Sinkhorn algorithm.

Since vanilla GANs thus seem to remain an interesting technique for inducing seed dictionaries, we explore what causes the instability of vanilla GAN seed induction, by looking at how they perform on simple transformations of the embedding spaces, and by using a combination of supervised training and model interpolation to analyze the loss landscapes. The results lead us to conclude that the instability is caused by a mild form of mode collapse, that cannot easily be overcome by changes in the number of parameters, batch size, and learning rate. Nevertheless, vanilla GANs with unsupervised model selection seem superior to more recently proposed methods, and we show that when combined with a state-of-the-art refinement technique, vanilla GANs with unsupervised model selection is superior to these methods across the board.

7.2 GAN-INITIALIZED UBDI

In this section, we discuss the dynamics of GAN-based UBDI. While the idea of using GANs for UBDI originates with Barone, (2016), we refer to Conneau et al., (2018) as the canonical implementation of GAN-based UBDI. Note that GANs are not a necessary component to unsupervised distribution matchning for alignment of vector spaces, albeit a popular approach (Barone, 2016; Conneau et al., 2018; Zhang et al., 2017b). In Section 7.3, we briefly discuss how GAN-based initialization compares to the alternative of using point set registration techniques (Hoshen and Wolf, 2018) and related strategies.

A GAN consists of a generator and a discriminator (Goodfellow et al., 2014). The generator G is trained to fool the discriminator D. The generator can be any differentiable function; in Conneau et al., (2018), it is a linear transform Ω. Let $\mathbf{e} \in E$ be an English word vector, and $\mathbf{f} \in F$ a French word vector, both of dimensionality d. The goal of the generator is then to choose $\Omega \in \mathbb{R}^{d \times d}$ such that ΩE has a distribution close to F. The discriminator is a map $D_w : \mathcal{X} \rightarrow \{0, 1\}$, implemented in Conneau et al., (2018) as a multi-layered perceptron. The objective of the discriminator is to discriminate between vector spaces F and ΩE. During training, the model parameters Ω and w are optimized using stochastic gradient descent by alternately updating the parameters of the discriminator based on the gradient of the discriminator loss and the parameters of the generator based on the gradient of the generator loss, which, by definition, is the inverse of the discriminator loss. The loss function used in Conneau et al., (2018) and in our experiments below is cross-entropy. In each iteration, we sample N vectors $e \in E$ and N vectors $f \in F$ and update the discriminator parameters w according to $w \rightarrow w + \alpha \sum_{i=1}^{N} \nabla [\log D_w(f_i) + \log(1 - D_w(G_\Omega(e_i)]$.

Theoretically, the optimal parameters are a solution to the min-max problem: $\min_\Omega \max_w \mathbb{E}[\log(D_w(F)) + \log(1 - D_w(G_\Omega(E)))]$, which reduces to $\min_\Omega JS(P_F \mid P_\Omega)$. If a generator wins the game against an ideal discriminator on a very large number of samples, then F and ΩE can be shown to be close in Jensen-Shannon divergence, and thus the model has learned the true data distribution. This result, referring to the distributions of the data, p_{data}, and the distribution, p_g, G is sampling from, is from Goodfellow et al., (2014): If G and D have enough capacity, and at each step of training, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion $E_{\mathbf{x} \sim p_{data}}[\log D^*_G(\mathbf{x})]$ then p_g converges to p_{data}. This result relies on a number of assumptions that do not hold in practice. The generator in Conneau et al., (2018), which learns a linear transform Ω, has very limited capacity, for example, and we are updating Ω rather than p_g. In practice, therefore, during training, Conneau et al., (2018) alternate between k steps of optimizing the discriminator and one step of optimizing the generator. Another common problem with training GANs is that the discriminator loss quickly drops to zero, when there is no overlap between p_g and p_{data} (Arjovsky, Chintala, and Bottou, 2017); but note that in our case, the discriminator is initially presented with IE and F, for which there is typically no trivial solution, since the embedding spaces are likely to overlap. We show in Section 7.4 the that discriminator and generator losses are poor model selection criteria, however; instead we propose a simple criterion based on cosine similarities between nearest neighbors in the learned alignment.

From ΩE and F, a seed (bilingual) dictionary can be extracted using nearest neighbor queries, i.e., by asking for the nearest neighbor of ΩE in F, or vice versa. Conneau et al., (2018) use a normalized

nearest neighbor retrieval method to reduce the influence of hubs (Dinu, Lazaridou, and Baroni, 2015; Radovanović, Nanopoulos, and Ivanovic, 2010). The method is called Cross-domain Similarity Local Scaling (CSLS) and used to expand high-density areas and condense low-density ones. The mean similarity of a source language embedding $\Omega\mathbf{e}$ to its k nearest neighbors in the target language is defined as $\mu_E^k(\Omega(\mathbf{e})) = \frac{1}{k}\sum_{i=1}^{k}\cos(\mathbf{e}, \mathbf{f}_i)$, where cos is the cosine similarity. $\mu_F(\mathbf{f}_i)$ is defined in an analogous manner for every i. $CSLS(\mathbf{e}, \mathbf{f}_i)$ is then calculated as $2\cos(\mathbf{e}, \mathbf{f}_i) - \mu_E(\Omega(\mathbf{e})) - \mu_F(\mathbf{f}_i)$. Conneau et al., (2018) use an unsupervised validation criterion based on CSLS. The translations of the top k (10,000) most frequent words in the source language are obtained with CSLS and average pairwise cosine similarity is computed over them. This metric is considered indicative of the closeness between the projected source space and the target space, and is found to correlate well with supervised evaluation metrics. After inducing a bilingual dictionary, E_d and F_d, by querying ΩE and F with CSLS, Conneau et al., (2018) perform a refinement step based on Procrustes Analysis (Schönemann, 1966). Here, the optimal mapping Ω that maps the words in the seed dictionary onto each other, is computed analytically as $\Omega = UV^T$, where U and V are obtained via the singular value decomposition $U\Sigma V^T$ of $F_d^T E_d$.

7.3 ALTERNATIVES TO GAN-INITIALIZED UBDI

This section introduces some recent alternatives to (vanilla) GAN-initialized UBDI. In Table 7.1, we list more approaches and classify them by how they perform unsupervised distribution matching and supervised refinement.

ITERATIVE CLOSEST POINT The idea of minimizing nearest neighbor distances for unsupervised model selection is also found in point set registration and lies at the core of ICP optimization (Besl and McKay, 1992). ICP typically minimizes the λ_2 distance (mean squared error) between nearest neighbor pairs. The ICP optimization algorithm works by assigning each transformed vector to its nearest neighbor and then computing the new relative transformation that minimizes the cost function with respect to this assignment. ICP can be shown to converge to local optima (Besl and McKay, 1992), in polynomial time (Ezra, Sharir, and Efrat, 2006). ICP easily gets trapped in local optima, however, exact algorithms only exist for two- and three-dimensional point set registration, and these algorithms are slow (Yang et al., 2016). Generally, it holds that the optimal solution to the GAN min-max problem is also optimal for ICP. To see this, note that a GAN minimizes the Jensen-Shannon divergence between F and ΩE. The optimal solution to this is $F = \Omega E$. As sample size goes to infinity, this means the $\mathcal{L}_2$ loss in ICP goes to 0. In other words, the ICP loss is minimal if

an optimal solution to the UBDI min-max problem is found. ICP was independently proposed for UBDI in Hoshen and Wolf, (2018). They report their method only works using PCA initialization, i.e. they project a subset of both sets of word embeddings onto the 50 first principal components, and learn an initial seed dictionary using ICP on the lower-dimensional embeddings. This seed mapping is then used as starting point for ICP on the full word embeddings. We explored PCA initialization for GAN-based distribution matching, but observed the opposite effect, namely that PCA initialization leads to a degradation in performance. The most important thing to note from Hoshen and Wolf, (2018), however, is that they do 500 random restarts of the PCA initialization to obtain robust performance; ICP, in other words, is extremely sensitive to initialization. This explains their poor performance under our experimental protocol below (Table 7.2).

WASSERSTEIN GAN Zhang et al., (2017b) were the first to introduce Wasserstein GANs as a way to learn seed dictionaries in the context of UBDI. In their best system, they train simple Wasserstein GANs and use the resulting seed dictionaries to supervise Procrustes Analysis. We modified the MUSE code to experiment with Wasserstein GANs in a controlled way. Simple Wasserstein GANs were unsuccessful, but with gradient penalty (Gulrajani et al., 2017), we obtained almost competitive results, after tuning the learning rate and the gradient penalty λ using nearest neighbor cosine distance as validation criterion. On the other hand, the results were not significantly better, and instability did not improve. Finally, we experimented with **CT-GAN!** (CT-GAN!)s (Wei et al., 2018), an extension of Wasserstein GANs with gradient penalty, but this only lowered performance and increased instability. Since Wasserstein GANs and CT-GANs were consistently worse and less stable than vanilla GANs, we do not include them in the experiments below.

GROMOV-WASSERSTEIN Alvarez-Melis and Jaakkola, (2018) present a very different initialization strategy. In brief, Alvarez-Melis and Jaakkola, (2018) learn a linear transformation to minimize Gromov-Wasserstein distances of distances between nearest neighbors, in the absence of cross-lingual supervision. We report the performance of their system in the experiments below, but results (Table 7.2) were all negative. We think the reason is that Alvarez-Melis and Jaakkola, (2018) only consider small subsamples of the vector spaces, and that in hard cases, alignments induced on subspaces are unlikely to scale. It achieved an impressive P@1 of 85.6 on the Greek MUSE dataset (Conneau et al., (2018) obtain 59.5); but on the datasets, where Conneau et al., (2018) are instable, considered here, it consistently fails to align the vector spaces.

Artetxe, Labaka, and Agirre, (2018a) introduce a very simple, related initialization method that is also based on Gromov-Wasserstein distances of distances between nearest neighbors: They use these second-order distances to build a seed dictionary directly by aligning nearest neighbors across languages. By itself, this is a poor initialization method (see Table 7.2). Artetxe, Labaka, and Agirre, (2018a), however, combine this with a new refinement method called *stochastic dictionary induction*, i.e., randomly dropping out dimensions of the similarity matrix when extracting a seed dictionary for the next iteration of Procrustes Analysis. Artetxe, Labaka, and Agirre, (2018a) show in an ablation study for one language pair (English-Finnish) that the initialization method only works in combination with the stochastic dictionary induction step, i.e., without the application of stochasticity, the induced mapping is degenerate. In our experiments below, we show that this finding generalizes to other language pairs, suggesting that the stochastic dictionary induction is the main contribution in their work. We show that when combined with vanilla GANs for the initial step of learning a seed dictionary through distribution matching, stochastic dictionary induction performs even better.

CONVEX RELAXATION The Gold-Rangarajan relaxation is a convex relaxation of the (NP-hard) graph matching problem and can be solved using the Frank-Wolfe algorithm. Once the minimal optimizer is computed, an initial transformation is obtained using singular-value decomposition. The Gold-Rangarajan relaxation can thus be used for stable learning of seed dictionaries Grave, Joulin, and Berthet, 2018. It remains an open question how this strategy fairs on challenging language pairs such as the ones included here. We would have liked to include this approach in our experiments, but the code was not publicly available at the time of writing.

PROPERTIES OF UNSUPERVISED ALIGNMENT ALGORITHMS The above approaches provably work if the two vector spaces to be aligned, are isomorphic, except for the pathological case where the vectors are placed on an equidistant grid forming a sphere.[1] A function Ω

1 In this case, there is an infinite set of equally good linear transformations (rotations) that achieve the same training loss. Similarly, for two binary-valued, n-dimensional vector spaces with one vector in each possible position. Here the number of local optima would be 2^n, but since the loss is the same in each of them the loss landscape is highly non-convex, and the basin of convergence is therefore very small (Yang et al., 2016). The chance of aligning the two spaces using gradient descent optimization would be $\frac{1}{2^n}$. In other words, minimizing the Jensen-Shannon divergence between the word vector distributions, even in the easy case, is not always guaranteed to uncover an alignment between translation equivalents. From the above, it follows that alignments between linearly alignable vector spaces cannot always be learned using UBDI methods. In Section **??** , we test for approximate isomorphism to decide whether two vector spaces are linearly alignable. Sections **??**–Section **??** are devoted to analyzing *when* alignments between linearly alignable vector spaces can be learned.

from E to F is a linear transformation if $\Omega(f+g) = \Omega(f) + \Omega(g)$ and $\Omega(kf) = k\Omega(f)$ for all elements f, g of E, and for all scalars k. An invertible linear transformation is called an *isomorphism*. The two vector spaces E and F are called isomorphic, if there is an isomorphism from E to F. Equivalently, if the kernel of a linear transformation between two vector spaces of the same dimensionality contains only the zero vector, it is invertible and hence an isomorphism. Most work on supervised or unsupervised alignment of word vector spaces relies on the assumption that they are approximately isomorphic, i.e., isomorphic after removing a small set of vertices (Barone, 2016; Conneau et al., 2018; Mikolov, Le, and Sutskever, 2013; Zhang et al., 2017b). It is not difficult to show that many pairs of vector spaces are not approximately isomorphic, however. See Søgaard, Ruder, and Vulić, 2018 for examples.

7.4 EXPERIMENTS

In our experiments, we focus on aligning word vector spaces between two languages, by projecting from the foreign language into English. Our languages are: Estonian (et), Farsi (fa), Finnish (fi), Latvian (lv), Turkish (tr), and Vietnamese (vi). This selection of languages is motivated by observed instability when training vanilla GANs, e.g., Søgaard, Ruder, and Vulić, 2018. In addition, the languages span four language families: Finno-Ugric (et, fi), Indo-European (fa, lv), Turkic (tr), and Austroasiatic (vi).

DATA In all our experiments, we use pretrained FastText embeddings (Bojanowski et al., 2017) and the bilingual dictionaries released along with the MUSE system.[2] The FastText embeddings are trained on Wikipedia dumps; the bilingual dictionaries were created using an in-house Facebook translation tool. Since we cannot do reliable hyperparameter optimization in the absence of cross-lingual supervision, we use MUSE with the default parameters (Conneau et al., 2018).

7.4.1 COMPARISON OF DISTRIBUTION MATCHING STRATEGIES

Our main experiments, reported in Table 7.2, compare the initialization strategies listed in Table 7.2: vanilla GANs, the two varieties of Gromov-Wasserstein (see Section 7.3), and ICP.[3] Table 7.2 is split in two: First we report the performance, measured as precision at one, in the absence of refinement; and then we report the performance *with* refinement, using *the same* refinement technique (Procrustes Analysis) across the

2 https://github.com/facebookresearch/MUSE

3 We ignore Wasserstein GANs, which proved more instable than vanilla GANs in our preliminary experiments, as well as Gold-Rangarajan, which performs considerably below current state of the art.

		to English													
		et		fa		fi		lv		tr		vi		**av**	
		max	fail	max	fail	max	fail	max	fail	max	fail	max	fail	max	fail
No Refinement															
Conneau et al., 2018	GAN	**6.4**	9	**22.5**	3	**28.5**	1	**14.3**	9	**32.1**	2	**2.4**	9	**17.1**	5.5
Hoshen and Wolf, 2018	ICP	0.1	10	0	10	0	10	0	10	0	10	0	10	0	10
Artetxe, Labaka, and Agirre, 2018a	GW	0	10	0.1	10	0.1	10	0.1	10	0.1	10	0.1	10	0.1	10
Alvarez-Melis and Jaakkola, 2018	GW	0	10	0	10	0	10	0	10	0	10	0	10	0	10
With Procrustes Refinement															
Conneau et al., 2018	GAN	**38.1**	9	**40.9**	3	58.9	1	**33.2**	9	**60.6**	2	**51.3**	9	**48.1**	5.5
Hoshen and Wolf, 2018	ICP	0.1	10	0	10	0	10	0	10	0	10	0	10	0	10
Artetxe, Labaka, and Agirre, 2018a	GW	1.1	10	40.2	0	**60.5**	0	0.1	10	59.6	0	0.3	10	27.0	5
Alvarez-Melis and Jaakkola, 2018	GW	0	10	0	10	0	10	0	10	0	10	0	10	0	10

Table 7.2: Comparisons of unsupervised **seed dictionary** learning strategies *in the absence of refinement* (upper half) or *using the same refinement technique* (orthogonal Procrustes) (lower half). For results with refinement, we use GANs, ICPs, and Gromov-Wasserstein (GW) distribution matching and feed seed dictionaries to Procrustes refinement. We then report maximum performance (P@1) and stability (fails) across 10 runs. We consider a P@1 score below 2% a failure. The results suggest that GANs, in spite of their instability, have the highest potential for inducing useful seed dictionaries.

board. For all the randomly initialized algorithms (the first three), we report the best of 10 runs and the number of *fails*, where fails are runs with scores lower than 2%. The reported scores are P@1, i.e., the fraction of words whose neighbors are translation equivalents.

We believe it is crucial to evaluate the different techniques this way, instead of simply comparing the numbers reported in the relevant papers: First of all, no three of these authors report performance on the same datasets. Secondly, if the authors use different refinement techniques, it is impossible to see the impact of the initialization strategies in the reported numbers. Instead we control for the refinement techniques and study the distribution matching techniques in Table 7.1 in isolation. This means, for example, that we evaluate the Artetxe, Labaka, and Agirre, (2018a) in the absence of stochastic dictionary induction, and Hoshen and Wolf, (2018) in the absence of 500 random restarts. In Section 7.4.2 (Table 7.3), we compare vanilla GANs and Gromov-Wasserstein in the context of stocastic dictionary induction.

The patterns in Table 7.2 are very consistent. Vanilla GAN distribution matching is very instable, with 1/10 fails for Finnish and Turkish, but 6, 7 and 9 fails for Estonian, Latvian, and Vietnamese, respectively. All other methods are *more* instable, however, with the distribution matching techniques in Hoshen and Wolf, 2018 and Alvarez-Melis and Jaakkola, 2018 failing across the board, with or without supervised Procrustes refinement. Vanilla GAN distribution matching also leads to higher precision for 5/6 language pairs.

Vanilla GAN distribution matching thus seems to have the highest potential for inducing useful seed dictionaries among all these methods. If we could only manage their instability, GANs seem to provide us with a better point of departure. This naturally leads us to ask: *Is it feasible to select good vanilla GAN UBDI runs from a batch of random restarts, in the absence of cross-lingual supervision?* This question is explored in Section 7.4.2, in which we also explore whether state-of-the-art performance can be achieved with vanilla GANs and a more advanced refinement technique, namely stochastic dictionary induction.

7.4.2 GAN DISTRIBUTION MATCHING WITH RANDOM RESTARTS

Exploring this question we found that the discriminator loss during training, which is used as a model selection criterion in Daskalakis et al., (2018), is a poor selection criterion. However, we did find another unsupervised model selection criterion that correlates well with UBDI performance: cosine similarity of (induced) nearest neighbors. This criterion is also used as a stopping criterion in Conneau et al., (2018), and can be used with or without CSLS scaling. This stopping criterion in fact turns out to be a quite robust model selection criterion for picking the best out of n random restarts.

	Procrustes	Stochastic Dictionary Induction	
	C-MUSE	C-MUSE	Artetxe, Labaka, and Agirre, (2018a)
et-en	38.1	47.6	47.6
fa-en	40.9	**41.5**	40.2
fi-en	58.9	62.5	**63.6**
lv-en	33.2	**44.6**	43.3
tr-en	60.6	**62.8**	60.6
vi-en	51.3	54.4	**55.3**
average	47.2	**52.2**	51.7

Table 7.3: Comparison of MUSE with cosine-based model selection over 10 random restarts (C-MUSE) with and without stochastic dictionary induction (with suggested hyper-parameters from Artetxe, Labaka, and Agirre, 2018a), against state of the art. Using vanilla GANs is slightly better than Gromov-Wasserstein on average and better on 3/6 language pairs.

In Table 7.3, we compare MUSE with 10 random restarts and using CSLS cosine similarity of nearest neighbors as an unsupervised model selection criterion, to the full state-of-the-art model in Artetxe, Labaka, and Agirre, (2018a) *with* stochastic dictionary induction. What we see in these results, is that Artetxe, Labaka, and Agirre, (2018a) is still superior to MUSE with random restarts, but even with 10 restarts, the gap narrows considerably, and MUSE is better on 2/6 languages. Note, however, that this is a comparison of two systems using two different refinement techniques. If we combine vanilla GAN distribution matching from MUSE with the stochastic dictionary induction technique from Artetxe, Labaka, and Agirre, (2018a), we obtain slightly better performance than Artetxe, Labaka, and Agirre, (2018a) (Table 7.3, mid-column): While overall improvements are small, compared to the differences in seed dictionary quality, the combination of vanilla GANs for distribution matching and stochastic dictionary induction provides a promising and fully competitive alternative to the state of the art for unsupervised word translation.

7.4.3 Discussion and further experiments

We have shown that while vanilla GANs are instable, they carry a seemingly unique potential for UBDI. We have shown that a simple unsupervised cosine-based model selection criterion can achieve robust state-of-the-art performance. We have performed several other experiments to probe this instability in search of ways to stabilize vanilla GANs without significant performance drops. This subsection summarizes these experiments.

NORMALIZATION We observed that GAN-based UBDI becomes more instable and performance deteriorates with unit length normalization. We performed unit length normalization (ULN) of all vectors $\mathbf{x}$, i.e., $\mathbf{x}' = \frac{\mathbf{x}}{||\mathbf{x}||^2}$, which is often used in supervised bilingual dictionary induction (Artetxe, Labaka, and Agirre, 2017; Xing et al., 2015). We used this transform to project word vectors onto a sphere – to control for shape information. If vectors are distributed smoothly over two spheres, there is no way to learn an alignment in the absence of dictionary seed; in other words, if vanilla GAN distribution matching is unaffected by this transform, vanilla GANs learn from density information alone. While supervised methods are insensitive to or benefit from ULN, we find that vanilla GANs are very sensitive to such normalization; in fact, the number of failed runs over six languages increases from below 50% to 90%. For example, while for Finnish, MUSE only fails in 1/10 runs, MUSE with ULN failed across the board; for Farsi, MUSE with ULN failed in 6/10 runs, compared to 3/10. We verify that supervised alignment is not affected by ULN by running Procrustes refinement with a seed dictionary as supervision; here, performance remains unchanged under this transformation.

NOISE INJECTION On the contrary, GAN-based UBDI is largely unaffected by noise injection. We saw this from running experiments on a few languages, but do not report performance across the board. Specifically, we add 25% random vectors, randomly sampled from a hypercube bounding the vector set. GAN-based UBDI results are not affected by noise injection. This, we found, is because the injected vectors rarely end up in the seed dictionaries used for subsequent refinement.

OVER-PARAMETERIZATION GAN training is instable because discriminators end up in poor local optima or saddle points (see below). A known technique for escaping local optima is over-parameterization (Brutzkus et al., 2018). We experimented with widening our discriminators to smoothen the loss landscape. Results were mixed, with more stability and better performance on some languages, and less stability and worse performance on others.

LARGE BATCHES AND SMALL LEARNING RATES Previous work has shown that large learning rate and small batch size contribute towards Stochastic Gradient Descent (SGD) finding flatter minima (Jastrzebski et al., 2018), but in our experiments, we are interested in the discriminator not ending up in flat regions, where there is no signal to update the generator. We therefore experiment with (higher and) *smaller* learning rate and (smaller and) *larger* batch sizes. The motivation behind both is decreasing the scale of random fluctuations in the SGD dynamics (Balles, Romero, and Hennig, 2017; Smith and

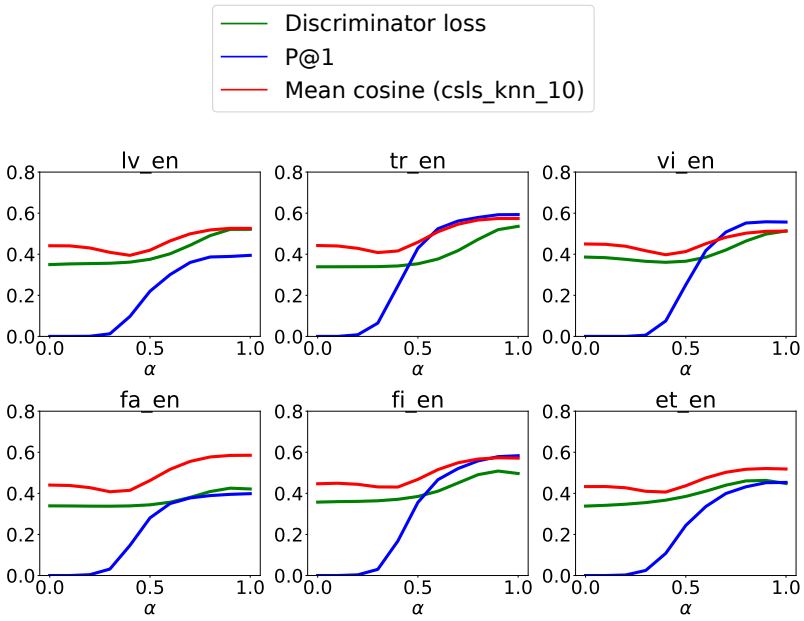

Figure 7.1: Discriminator loss averaged over all training data points (green), P@1 on the test data points (blue) and mean cosine similarity (red) on the training data – for generator parameters on the line segment that connects the unsupervised GAN solution with the supervised Procrustes Analysis solution. α is the interpolation parameter moving the generator parameters from the unsupervised GAN solution ($\alpha = 0$) to the supervised solution ($\alpha = 1$).

Le, 2017), enabling the discriminator to explore narrower regions in the loss landscape. Increasing the batch size or varying the learning rate (up or down), however, leads to worse performance, and it seems the MUSE default hyperparameters are close to optimal.

EXPLORING THE LOSS LANDSCAPES GAN training instability arises from discriminators getting stuck in saddle points, where neither the discriminator nor the generator has a learning signals. To show this, we analyze the discriminator loss in areas of convergence by plotting it as a function of the generator parameters. Specifically, we plot the loss surface along its intersection with a line segment connecting two sets of parameters (Goodfellow, Vinyals, and Saxe, 2015; Li et al., 2018). In our case, we interpolate between the model induced by GAN-based UBDI and the (oracle) model obtained using supervised Procrustes Analysis. Results are shown in Figure 7.1. The green loss curves represent the current discriminator's loss along all the generators between the current generator and the generator found by Procrustes refinement. We see that while performance (P@1 and mean cosine similarity) goes up as soon as we move closer toward the supervised solution, the discriminator loss does not change until we get very close to this solution, suggesting there is no learning signal in this direction

for GAN-based UBDI. This is along a line segment representing the shortest path from the failed generator to the oracle generator, of course; linear interpolation provides no guarantee there are no almost-as-short paths with plenty of signal. A more sophisticated sampling method is to sample along two random direction vectors (Goodfellow, Vinyals, and Saxe, 2015; Li et al., 2018). We used an alternative strategy of sampling from normal distributions with fixed variance that were orthogonal to the line segment. We observed the same pattern, leading us to the conclusion that instability is caused by discriminator saddle points.

7.5 CONCLUSIONS

This paper explores the dynamics of (vanilla) GAN training in the context of unsupervised word translation and a systematic comparison of GANs with different distribution matching (seed induction) methods across six challenging language pairs. Our main finding is that vanilla GANs, in spite of their instability, have the highest potential for inducing useful seed dictionaries. We explore an unsupervised model selection criterion for selecting the best models from multiple random restarts, narrowing the gap between MUSE and Artetxe, Labaka, and Agirre, 2018a, and further show that combining GANs with stochastic dictionary induction provides a new state of the art for unsupervised word translation.

Part IV

CONCLUSION

8

DISCUSSION OF THE CONTRIBUTIONS

The preceding chapters presented research into automatic content coding for text data from different domains or languages. As manual annotation of datasets for content coding is expensive, cross-domain and cross-language transfer should help automatic content coding to leverage information from existing datasets. The first research question asked in this thesis was

Can transfer learning be useful for the automatic coding of content?

The work presented in this thesis confirms that this is indeed the case. In Chapter 4, we showed that multi-task learning enables the coding of posts from online discussion fora, a domain for which no labeled training data exists. The multi-task approach solves this problem by enabling the exploitation of three types of readily available resources and does not require any additional coding of content, besides a small dataset for evaluation. The additional resources included a large frame-labeled dataset of news articles, large amounts of unlabeled text from online discussion fora, and a smaller set of additional annotations of the discussion posts with labels that are not directly relevant to the frame-labeling task. As it seems reasonable to expect the availability of such resources for other content coding problems, multi-task learning proves an ideal approach for extending content coding to new domains. In Chapter 3, we attempted to overcome the small size of the training data set with cross-lingual transfer using cross-lingual word embeddings. In that case, the cross-lingual transfer did not improve results. However, the experiments are carried out in a distant supervision setup, i.e. the additional Russian data is expected to be of the pro-Russian target class, but this expectation is not confirmed by manually labeling the data. Hence, the lack of performance improvement might be due to noise in the additional data Hence, we do not take this experiment as evidence that transfer learning does *not* help in general, but that it does not help in this specific case. An interesting question for future work is to explore under which conditions transfer learning does help the content coding task, and how to best select which kind of additional data to leverage for improving model performance.

The second part of the thesis focused on methods that enable cross-lingual transfer to answer the second research question:

How can we improve word representations that capture semantics across languages?

In Chapter 5, we evaluated an approach for unsupervised bilingual dictionary induction from image data on a dataset that besides nouns also contained verbs and adjectives. We found that translations can be induced reliably for concrete nouns, but the model struggles with translating verbs and adjectives. This shows that in order to assess model performance for inducing word-level translations, the choice of evaluation data is crucial, as performance for words with specific features might not generalize to other groups of words.

In Chapters 6 and 7, we analysed training instabilities of the GAN-based MUSE system for unsupervised word embedding alignment.

In Chapter 6, we showed that the system cannot align two sets of English embeddings learned by different embedding algorithms. In Chapter 7, we further investigated the unsupervised alignment between English and another language, and found that the training instabilities are most likely caused by saddle points in which the discriminator gets stuck without a learning signal. We also found that this problem cannot be easily overcome by varying hyperparameters such as batch size and learning rate, or overparameterization of the discriminator. This is unfortunate, as our comparison between several systems for unsupervised alignment revealed that in a successful run, the MUSE system has the highest potential to induce useful starting points for subsequent iterative refinement. Finally, we showed that such successful runs can be selected using an unsupervised criterion based on cosine similarity.

Even though this selection criterion can tell us which initial seed dictionaries to select for subsequent iterative refinement, it does not alleviate the training instabilities. How these instabilities can be overcome is still an open problem. In a recent study, Vulić et al., 2019 found that even the empirically most stable unsupervised alignment system (VecMap) fails in many configurations. They point out that especially unsupervised methods suffer from the fact that the isomorphism assumption does not hold (Søgaard, Ruder, and Vulić, 2018), and conclude that one of the most promising research directions for embedding alignment systems, regardless of the degree of supervision, is to increase the isomorphism between the monolingual embedding spaces. The conclusion of the presented thesis aligns with that suggestion, and our future work will be aimed at increasing isomorphism between embedding spaces.

Part V

APPENDIX

9 APPENDIX

9.1 DATA PREPROCESSING

For the Twitter and news articles datasets, we remove all instances that do not correspond to the five target frames. Table **??** shows the class distributions in the filtered datasets. We tokenize all sequences using spaCy [1], which we also use for sentence splitting in the news articles dataset. For the Twitter dataset, we follow Johnson, Jin, and Goldwasser, (2017) in removing URLs and @-mentions.

9.2 HYPERPARAMETERS IN EXPERIMENTS

The hyperparameters for all neural models were tuned on the online disc. dev set. We report test results for the optimal settings found by averaging over 3 training runs, which we determine by the best macro-averaged F-score and smallest variance between the runs. We set the DyNet weight decay parameter to 1e-7 for all neural models, batch size is 128, and the word embeddings are not updated during training.

For the multi-task and adversarial model, we do a grid-search over the weight of the coin flip used to decide on sampling from main/aux or main/adversarial task in the range of [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]. The optimal weight for sampling the main task is 0.5 for the multi-task model and 0.3 for the adversarial task.

All models are trained using early stopping (after at least 80 epochs of training) with a patience of 5 epochs. The number of iterations (updates) per epoch is a hyperparameter, that we set by default as twice the number of data batches for the main task. For a fair coin flip, the models hence see as much data for the main task as for the auxiliary/adversarial task per epoch.

1 https://spacy.io/

Dataset	# instances	# instances per class					# multi
		1	13	5	6	7	
Newspaper (Train)	10,480	1088	1959	2023	924	890	45
Twitter (Train)	1,636	73	300	137	27	174	554
Online Disc. (Test)	692	78	96	234	166	186	67
			0	1			
Argument Quality	3,785		1,350	2,435			0
Online Disc. unlabeled	4731						

Table 9.1: Dataset statistics and class distributions. The frame labels correspond to the classes *Economic* (1), *Political* (13), *Legality, Jurisprudence and Constitutionality* (5), *Policy prescription and evaluation* (6) and *Crime and Punishment* (7). # *multi* refers to the number of multi-label instances. For Argument quality, label 1 indicates a score greater or equal 0.5.

References

Paul D Almeida and Mark Irving Lichbach. 2003. To the internet, from the internet: Comparative media coverage of transnational protests. *Mobilization: An International Journal*, 8(3):249–272.

Héctor Martínez Alonso and Barbara Plank. 2016. When is multitask learning effective? semantic sequence prediction under varying data conditions. *arXiv preprint arXiv:1612.02251*.

David Alvarez-Melis and Tommi Jaakkola. 2018. Gromov-wasserstein alignment of word embedding spaces. In *EMNLP*.

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016. Massively multilingual word embeddings. *CoRR*, abs/1602.01925.

Matheus Araujo, Julio Reis, Adriano Pereira, and Fabricio Benevenuto. 2016. An evaluation of machine translation for multilingual sentence-level sentiment analysis. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 1140–1145. ACM.

Martin Arjovsky and Leon Bottou. 2017. Towards principled methods for training generative adversarial networks. In *ICLR*.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. In *CoRR*, page abs/1701.07875.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of AAAI*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *ACL*.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance Detection with Bidirectional Conditional Encoding. In *Proceedings of EMNLP*.

Isabelle Augenstein, Sebastian Ruder, and Anders Søgaard. 2018. Multi-Task Learning of Pairwise Sequence Classification Tasks over Disparate Label Spaces. In *NAACL-HLT*, pages 1896–1906. Association for Computational Linguistics.

Isabelle Augenstein and Anders Søgaard. 2017. Multi-Task Learning of Keyphrase Boundary Classification. In *Proceedings of ACL*.

Mitja D Back, Albrecht C P Küfner, and Boris Egloff. 2010. The Emotional Timeline of September 11, 2001. *Psychological Science*, 21(10):1417–1419.

Mitja D Back, Albrecht CP Küfner, and Boris Egloff. 2011. " automatic or the people?": Anger on september 11, 2001, and lessons learned for the analysis of large digital data sets. *Psychological Science*, 22(6):837.

Akshat Bakliwal, Jennifer Foster, Jennifer van der Puil, Ron O'Brien, Lamia Tounsi, and Mark Hughes. 2013. Sentiment Analysis of Political Tweets: Towards an Accurate Classifier. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 49–58, Atlanta, Georgia. Association for Computational Linguistics.

Alexandra Balahur and Marco Turchi. 2012. Multilingual sentiment analysis using machine translation? In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '12, pages 52–60, Stroudsburg, PA, USA. Association for Computational Linguistics.

Georgios Balikas, Simon Moura, and Massih-Reza Amini. 2017. Multitask learning for fine-grained twitter sentiment analysis. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 1005–1008. ACM.

Lukas Balles, Javier Romero, and Philipp Hennig. 2017. Coupling adaptive batch sizes with learning rates. In *Proceedings of UAI*.

Antonio Valerio Miceli Barone. 2016. Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders. *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 121–126.

Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. 2009. Gephi: An open source software for exploring and manipulating networks.

Eric Baumer, Elisha Elovic, Ying Qin, Francesca Polletta, and Geri Gay. 2015. Testing and Comparing Computational Approaches for Identifying the Language of Framing in Political News. In *Proceedings of HLT-NAACL*, pages 1472–1482. The Association for Computational Linguistics.

Mikhail Bautin, Lohit Vijayarenu, and Steven Skiena. 2008. International sentiment analysis for news and blogs. In *ICWSM*.

Shane Bergsma and Randy Goebel. 2011. Using Visual Information to Predict Lexical Preference. In *Proceedings of RANLP*, RANLP '11, Hissar, Bulgaria.

Shane Bergsma and Benjamin Van Durme. 2011. Learning Bilingual Lexicons using the Visual Similarity of Labeled Web Images. In *Proceedings of IJCAI*, IJCAI '11, Barcelona, Spain.

Paul Besl and Neil McKay. 1992. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2).

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of EACL*.

Johannes Bjerva. 2017. Will my auxiliary tagging task help? estimating auxiliary tasks effectivity in multi-task learning. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*, 131, pages 216–220. Linköping University Electronic Press.

David M Blei and John D Lafferty. 2009. Topic models. In *Text Mining*, pages 101–124. Chapman and Hall/CRC.

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

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

Amber E. Boydstun, Dallas Card, Justin H. Gross, Philip Resnik, and Noah A. Smith. 2014. Tracking the Development of Media Frames within and across Policy Issues. In *Proceedings of APSA*.

Daniela Braun, Maike Salzwedel, Christian Stumpf, and Andreas M Wüst. 2007. Euromanifesto documentation. *Mannheim: MZES*.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49(1):1–47.

Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. 2018. SGD learns over-parameterized networks that provably generalize on linearly separable data. In *Proceedings of ICLR*.

Ceren Budak and Duncan J. Watts. 2015. Dissecting the spirit of gezi: Influence vs. selection in the occupy gezi movement. *Sociological Science*, 2(18):370–397.

Björn Burscher, Daan Odijk, Rens Vliegenthart, Maarten de Rijke, and Claes H de Vreese. 2014. Teaching the Computer to Code Frames in News: Comparing Two Supervised Machine Learning Approaches to Frame Analysis. *Communication Methods and Measures*, 8(3):190–206.

Dallas Card, Amber E Boydstun, Justin H Gross, Philip Resnik, and Noah A Smith. 2015. The Media Frames Corpus: Annotations of Frames Across Issues. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 438–444, Beijing, China. Association for Computational Linguistics.

Dallas Card, Justin Gross, Amber Boydstun, and Noah A Smith. 2016. Analyzing Framing through the Casts of Characters in the News. In *Proceedings of EMNLP*, pages 1410–1420.

Richard Caruana. 1993. Multitask Learning: A Knowledge-Based Source of Inductive Bias. In *Proceedings of ICML*, pages 41–48. Morgan Kaufmann.

Noam Chomsky and Edward Herman. 1988. Manufacturing Consent New York. *Pantheon*.

Dennis Chong and James Druckman. 2007. Framing Theory. *Annual Review of Political Science*, 10.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *JMLR*, 999888:2493–2537.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of ICLR*.

Michael D Conover, Emilio Ferrara, Filippo Menczer, and Alessandro Flammini. 2013. The digital evolution of occupy wall street. *PloS one*, 8(5):e64679.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Frank E. Dardis, Frank R. Baumgartner, Amber E. Boydstun, Suzanna de Boef, and Fuyuan Shen. 2008. Media Framing of Capital Punishment and Its Impact on Individuals' Cognitive Responses. *Mass Communication and Society*, 11(2):115–140.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of ACL*, pages 600–609.

Kia Dashtipour, Soujanya Poria, Amir Hussain, Erik Cambria, Ahmad YA Hawalah, Alexander Gelbukh, and Qiang Zhou. 2016. Multilingual sentiment analysis: state of the art and independent comparison of techniques. *Cognitive computation*, 8(4):757–771.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. 2018. Training GANs with optimism. In *ICLR*.

Dina Demner-Fushman and Douglas W Oard. 2003. The effect of bilingual term list size on dictionary-based cross-language information retrieval. In *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, pages 10–pp. IEEE.

Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. 2016. Tweet2Vec: Character-Based Distributed Representations for Social Media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 269–274, Berlin, Germany. Association for Computational Linguistics.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. In *Proceedings of ICLR (Workshop Track)*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-Task Learning for Multiple Language Translation. In *Proceedings of ACL*.

Susan T Dumais, Thomas K Landauer, and Michael L Littman. 1996. Automatic cross-linguistic information retrieval using latent semantic indexing. In *Working Notes of the Workshop on Cross-Linguistic Information Retrieval. ACM SIGIR*.

Kathleen T Durant and Michael D Smith. 2006. Predicting the political sentiment of web log posts using supervised machine learning techniques coupled with feature selection. In *International Workshop on Knowledge Discovery on the Web*, pages 187–206. Springer.

Greg Durrett, Adam Pauls, and Dan Klein. 2012. Syntactic transfer using a bilingual lexicon. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1–11. Association for Computational Linguistics.

Miles Efron. 2010. Hashtag retrieval in a microblogging environment. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 787–788. ACM.

Robert M. Entman. 1993. Framing: Toward Clarification of a Fractured Paradigm. *Journal of Communication*, 43(4):51–58.

Michael Evans, Wayne McIntosh, Jimmy Lin, and Cynthia Cates. 2007. Recounting the courts? applying automated content analysis to enhance empirical legal research. *Journal of Empirical Legal Studies*, 4(4):1007–1039.

Ester Ezra, Micha Sharir, and Alon Efrat. 2006. On the ICP algorithm. In *SGC*.

Don Fallis. 2015. What is disinformation? *library trends*, 63(3):401–426.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of EACL*, pages 462–471.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of REPEVAL*, pages 30–35.

William Ferreira and Andreas Vlachos. 2016. Emergent: A Novel Data-Set for Stance Classification. In *Proceedings of NAACL HLT*.

Anjalie Field, Doron Kliger, Shuly Wintner, Jennifer Pan, Dan Jurafsky, and Yulia Tsvetkov. 2018. Framing and Agenda-setting in Russian News: a Computational Analysis of Intricate Political Strategies. In *Proceedings of EMNLP*, pages 3570–3580. Association for Computational Linguistics.

Blaz Fortuna and John Shawe-Taylor. 2005. The use of machine translation tools for cross-lingual text mining. In *Proceedings of the ICML Workshop on Learning with Multiple Views*.

Pascale Fung. 1995. Compiling bilingual lexicon entries from a non-parallel english-chinese corpus. In *Third Workshop on Very Large Corpora*.

John D. Gallacher, Vlad Barash, Philip N. Howard, and John Kelly. 2018. Junk news on military affairs and national security: Social media disinformation campaigns against us military personnel and veterans. *ArXiv*, abs/1802.03572.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France. PMLR.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity. In *Proceedings of EMNLP*.

Goran Glavaš, Marc Franco-Salvador, Simone P Ponzetto, and Paolo Rosso. 2018. A resource-light method for cross-lingual semantic textual similarity. *Knowledge-Based Systems*, 143:1–9.

Goran Glavas, Robert Litschko, Sebastian Ruder, and Ivan Vulic. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. *arXiv preprint arXiv:1902.00508*.

Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. 2013. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 593–596. ACM.

S Gold and A Rangarajan. 1996. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:377–388.

Yevgeniy Golovchenko, Mareike Hartmann, and Rebecca Adler-Nissen. 2018. State, media and civil society in the information warfare over Ukraine: citizen curators of digital disinformation. *International Affairs*, 94(5):975–994.

Sonia Maria Guedes Gondim and Pedro Fernando Bendassolli. 2014. The use of the qualitative content analysis in psychology: A critical review. *Psicologia em Estudo*, 19(2):191–199.

Sandra González-Bailón, Javier Borge-Holthoefer, and Yamir Moreno. 2013. Broadcasters and hidden influentials in online protest diffusion. *American Behavioral Scientist*, 57(7):943–965.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David WardeFarley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. In *Proceedings of NIPS*.

Ian J Goodfellow, Oriol Vinyals, and Andrew Saxe. 2015. Qualitatively characterizing neural network optimization problems. In *Proceedings of ICLR*.

Stephan Gouws and Anders Søgaard. 2015. Simple task-specific bilingual word embeddings. In *Proceedings of NAACL-HLT*, pages 1302–1306.

Edouard Grave, Armand Joulin, and Quentin Berthet. 2018. Unsupervised alignment of embeddings with wasserstein procrustes. *CoRR*, abs/1805.11222.

Gregory Grefenstette. 1998. Evaluating the adequancy of a multilingual transfer dictionary for the cross language information retrieval. LREC.

Justin Grimmer. 2010. A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Political Analysis*, 18(1):1–35.

Justin Grimmer and Brandon M Stewart. 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political analysis*, 21(3):267–297.

Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire, and David Lazer. 2019. Fake news on twitter during the 2016 u.s. presidential election. *Science*, 363:374–378.

Andrew Guess, Jonathan Nagler, and Joshua Tucker. 2019. Less than you think: Prevalence and predictors of fake news dissemination on facebook. In *Science advances*.

Harold Guetzkow. 1959. Propaganda analysis: A study of inferences made from nazi propaganda in world war ii. by alexander l. george. (evanston, ill., white plains, n. y.: Row, peterson and co.1959. pp. xxii, 287. $6.00.). *American Political Science Review*, 53(4):1129–1131.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of Wasserstein GANSs. In *Proceedings of NIPS*.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. 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)*, pages 1234–1244.

Alexander Hanna. 2013. Computer-aided content analysis of digitally enabled movements. *Mobilization: An International Quarterly*, 18(4):367–388.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M Meyer, and Iryna Gurevych. 2018. A Retrospective Analysis of the Fake News Challenge Stance-Detection Task. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1859–1874, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Like Harding. 2019. Three russians and one ukrainian to face mh17 murder charges. *The Guardian*.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Mareike Hartmann, Yevgeniy Golovchenko, and Isabelle Augenstein. 2019a. Mapping (dis)information about the mh17 plane crash on twitter. In *To appear in Proceedings of the 2nd Workshop on NLP for Internet Freedom (NLP4IF)*.

Mareike Hartmann, Tallulah Jansen, Isabelle Augenstein, and Anders Søgaard. 2019b. Issue Framing in Online Discussion Fora. 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)*, pages 1401–1407.

Mareike Hartmann, Yova Kementchedjhieva, and Anders Søgaard. 2018. Why is unsupervised alignment of english embeddings from different algorithms so hard? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mareike Hartmann, Yova Kementchedjhieva, and Anders Søgaard. 2019c. Comparing unsupervised word translation methods step by step. In *To appear in Proceedings of NeurIPS*.

Mareike Hartmann and Anders Søgaard. 2018. Limitations of cross-lingual learning from image search. In *Proceedings of The Third Workshop on Representation Learning for NLP (Repl4NLP)*, pages 159–163.

Kazi Saidul Hasan and Vincent Ng. 2013. Stance Classification of Ideological Debates: Data, Models, Features, and Constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1348–1356, Nagoya, Japan. Asian Federation of Natural Language Processing.

Bradley Hauer, Garrett Nicolai, and Grzegorz Kondrak. 2017. Bootstrapping unsupervised bilingual lexicon induction. In *Proceedings of EACL*, pages 619–624.

Marti A. Hearst. 1999. Untangling text data mining. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 3–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Felix Hill and Anna Korhonen. 2014. Learning abstract concept embeddings from multi-modal data: Since you probably can't see what I mean. In *Proceedings of EMNLP*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *CoRR*, abs/1408.3456.

Dustin Hillard, Stephen Purpura, and John Wilkerson. 2008. Computer-assisted topic classification for mixed-methods social science research. *Journal of Information Technology & Politics*, 4(4):31–46.

Frederik Hjorth and Rebecca Adler-Nissen. 2019. Ideological Asymmetry in the Reach of Pro-Russian Digital Disinformation to United States Audiences. *Journal of Communication*, 69(2):168–192.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural Computation*, 9.

David L Hoover. 2008. Quantitative analysis and literary studies. *A Companion to Digital Literary Studies*, pages 517–533.

Daniel J Hopkins and Gary King. 2010. A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, 54(1):229–247.

Yedid Hoshen and Lior Wolf. 2018. An iterative closest point method for unsupervised word translation. In *CoRR*, page 1801.06126.

Philip N Howard, Bharath Ganesh, Dimitra Liotsiou, John Kelly, and Camille François. 2018. *The IRA, social media and political polarization in the United States, 2012-2018*. University of Oxford.

Hsiu-Fang Hsieh and Sarah E Shannon. 2005. Three approaches to qualitative content analysis. *Qualitative health research*, 15(9):1277–1288.

Shu Huang, Wei Peng, Jingxuan Li, and Dongwon Lee. 2013. Sentiment and topic analysis on social media: a multi-task multi-label classification approach. In *Proceedings of the 5th annual ACM web science conference*, pages 172–181. ACM.

Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. 2013. Extracting information nuggets from disaster-related messages in social media. In *Iscram*.

Ann Irvine and Chris Callison-Burch. 2017. A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

Shanto Iyengar. 1991. *Is Anyone Responsible? How Television Frames Political Issues*. University of Chicago Press.

Stanislaw Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. 2018. Finding flatter minima with SGD. In *Proceedings of ICLR*.

Yangfeng Ji and Noah Smith. 2017. Neural Discourse Structure for Text Categorization. In *Proceedings of ACL*.

Christopher Johnson, Parul Shukla, and Shilpa Shukla. 2012. On classifying the political sentiment of tweets. *Cs. utexas. edu*.

Kristen Johnson, Di Jin, and Dan Goldwasser. 2017. Leveraging Behavioral and Social Information for Weakly Supervised Collective Classification of Political Discourse on Twitter. In *Proceedings of ACL*.

Aditya Joshi, AR Balamurali, and Pushpak Bhattacharyya. 2010. A fall-back strategy for sentiment analysis in hindi: a case study. *Proceedings of the 8th ICON*.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Garth S. Jowett and Victoria O'donnell. 2014. *Propaganda & persuasion*. Sage.

John Kelly, Vladimir Barash, Karina Alexanyan, Bruce Etling, Robert Faris, Urs Gasser, and John G Palfrey. 2012. Mapping Russian Twitter. *Berkman Center Research Publication*, (2012-3).

Yova Kementchedjhieva, Mareike Hartmann, and Anders Søgaard. 2019. Lost in evaluation: Misleading benchmarks for bilingual dictionary induction. In *To appear in Proceedings of EMNLP*.

Douwe Kiela and Léon Bottou. 2014. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of EMNLP*.

Douwe Kiela, Felix Hill, Anna Korhonen, and Stephen Clark. 2014. Improving multi-modal representations using image dispersion: Why less is sometimes more. In *Proceedings of ACL*, Baltimore, Maryland.

Douwe Kiela, Anita Lilla Verő, and Stephen Clark. 2016. Comparing data sources and architectures for deep visual representation learning in semantics. In *Proceedings of EMNLP*.

Douwe Kiela, Ivan Vulic, and Stephen Clark. 2015. Visual bilingual lexicon induction with transferred convnet features. In *Proceedings of EMNLP*.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Gary King and Will Lowe. 2003. An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. *International Organization*, 57(3):617–642.

Gary King, Jennifer Pan, and Margaret E Roberts. 2013. How censorship in china allows government criticism but silences collective expression. *American Political Science Review*, 107(2):326–343.

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

Sergei Koltcov, Olessia Koltsova, and Sergey Nikolenko. 2014. Latent dirichlet allocation: Stability and applications to studies of user-generated content. In *Proceedings of the 2014 ACM Conference on Web Science*, WebSci '14, pages 161–165, New York, NY, USA. ACM.

Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of NIPS*.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *Proceedings of ICLR (Conference Papers)*.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Harold D Lasswell. 1948. The structure and function of communication in society. *The communication of ideas*, 37(1):136–39.

Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. 2015. Combining language and vision with a multimodal skip-gram model. In *Proceedings of NAACL HLT*.

Matthias Lemke, Andreas Niekler, Gary S Schaal, and Gregor Wiedemann. 2015. Content Analysis between Quality and Quantity. *Datenbank-Spektrum*, 15(1):7–14.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proceedings of NIPS*, pages 2177–2185.

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

Omer Levy, Yoav Goldberg, and Anders Søgaard. 2017. A strong baseline for learning cross-lingual word embeddings from sentence alignments. In *Proceedings of EACL*.

Hao Li, Zheng Xu, Gavin Taylor, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. In *ICLR*.

Chenghua Lin, Yulan He, and Richard Everson. 2011. Sentence Subjectivity Detection with Weakly-Supervised Learning. In *Proceedings of IJCNLP*, pages 1153–1161, Chiang Mai, Thailand.

Ying Lin, Shengqi Yang, Veselin Stoyanov, and Heng Ji. 2018. A Multi-lingual Multi-task Architecture for Low-resource Sequence Labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 799–809, Melbourne, Australia. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742*.

Pintu Lohar, Haithem Afli, and Andy Way. 2017. Maintaining sentiment polarity in translation of user-generated content. *The Prague Bulletin of Mathematical Linguistics*, 108(1):73–84.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Detect rumor and stance jointly by neural multi-task learning. In *Companion Proceedings of the The Web Conference 2018*, pages 585–593. International World Wide Web Conferences Steering Committee.

Walid Magdy, Kareem Darwish, and Ingmar Weber. 2016. # failedrevolutions: Using twitter to study the antecedents of isis support. In *2016 AAAI Spring Symposium Series*.

Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. 2010. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103.

Elijah Meeks and Scott B Weingart. 2012. The digital humanities contribution to topic modeling. *Journal of Digital Humanities*, 2(1):1–6.

I Dan Melamed. 1995. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. *arXiv preprint cmp-lg/9505044*.

Nicolas Merz, Sven Regel, and Jirka Lewandowski. 2016. The manifesto corpus: A new resource for research on political parties and quantitative text analysis. *Research & Politics*, 3(2):2053168016643346.

Rada Mihalcea, Carmen Banea, and Janyce Wiebe. 2007. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 976–983.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS*.

Tomas Mikolov, Kai Chen, Gregroy S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

David Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *Proceedings of EMNLP*.

David Mimno, Hanna M Wallach, Jason Naradowsky, David A Smith, and Andrew McCallum. 2009. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 880–889. Association for Computational Linguistics.

Jeff Mitchell and Mark Steedman. 2015. Orthogonality of syntax and semantics within distributional spaces. In *Proceedings of ACL*.

Aditya Mogadala and Achim Rettinger. 2016. Bilingual word embeddings from parallel and non-parallel corpora for cross-language text classification. In *Proceedings of NAACL-HLT*, pages 692–702.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016a. SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of SemEval*.

Saif M Mohammad, Mohammad Salameh, and Svetlana Kiritchenko. 2016b. How translation alters sentiment. *Journal of Artificial Intelligence Research*, 55:95–130.

Fred Morstatter, Liang Wu, Uraz Yavanoglu, Stephen R Corman, and Huan Liu. 2018. Identifying framing bias in online news. *ACM Transactions on Social Computing*, 1(2):5.

Nona Naderi and Graeme Hirst. 2017. Classifying Frames at the Sentence Level in News Articles. In *Proceedings of RANLP*, pages 536–542.

Laura K Nelson, Derek Burk, Marcel Knudsen, and Leslie McCall. 2018. The future of coding: a comparison of hand-coding and three types of computer-assisted text analysis methods. *Sociological Methods & Research*, page 0049124118769114.

Sarah Oates. 2016. Russian media in the digital age: Propaganda rewired. *Russian Politics*, 1(4):398–417.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Arash Heydarian Pashakhanlou. 2017. Fully integrated content analysis in International Relations. *International Relations*, 31(4):447–465.

Michael J Paul and Mark Dredze. 2011. You are what you tweet: Analyzing twitter for public health. In *Fifth International AAAI Conference on Weblogs and Social Media*.

Bolette Pedersen, Sanni Nimb, Anders Søgaard, Mareike Hartmann, and Sussi Olsen. 2018. A danish framenet lexicon and an annotated corpus used for training and evaluating a semantic frame classifier. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of EMNLP*.

Senja Pollak, Roel Coesemans, Walter Daelemans, and Nada Lavrač. 2011. Detecting contrast patterns in newspaper articles by combining discourse analysis and text mining. *Pragmatics. Quarterly Publication of the International Pragmatics Association (IPrA)*, 21(4):647–683.

Peter Pomerantsev and Michael Weiss. 2014. *The menace of unreality: How the Kremlin weaponizes information, culture and money*.

Stephen Purpura and Dustin Hillard. 2006. Automated classification of congressional legislation. In *Proceedings of the 2006 International Conference on Digital Government Research*, dg.o '06, pages 219–225. Digital Government Society of North America.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D Manning. 2018. Universal Dependency Parsing from Scratch. In *Proceedings of the {CoNLL} 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Milos Radovanović, Alexandros Nanopoulos, and Mirjana Ivanovic. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. *arXiv preprint cmp-lg/9505037*.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Daniel Jurafsky. 2013. Linguistic Models for Analyzing and Detecting Biased Language. In *Proceedings of ACL*.

Marek Rei. 2017. Semi-supervised Multitask Learning for Sequence Labeling. In *Proceedings of ACL*.

Richard Rogers. 2017. Digital methods for cross-platform analysis. *The SAGE handbook of social media*, pages 91–110.

Sebastian Ruder. 2019. *Neural Transfer Learning for Natural Language Processing*. Ph.D. thesis, National University of Ireland, Galway.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2019a. Multi-Task Architecture Learning. In *AAAI*.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019b. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3).

Takuto Sakamoto and Hiroki Takikawa. 2017. Cross-national measurement of polarization in political discourse: Analyzing floor debate in the us the japanese legislatures. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3104–3110. IEEE.

Matthew Salganik. 2019. *Bit by bit: Social research in the digital age*. Princeton University Press.

Cícero dos Santos and Maíra Gatti. 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Charles Schafer and David Yarowsky. 2002. Inducing translation lexicons via diverse similarity measures and bridge languages. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–7. Association for Computational Linguistics.

Michael Scharkow. 2013. Thematic content analysis using supervised machine learning: An empirical evaluation using german online news. *Quality & Quantity*, 47:761–773.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10.

Peter Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10.

Philip A Schrodt, Shannon G Davis, and Judith L Weddle. 1994. Political science: Keds—a program for the machine coding of event data. *Social Science Computer Review*, 12(4):561–587.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Stephen B. Seidman. 1983. Network structure and minimum degree. *Social Networks*, 5(3):269 – 287.

Qinlan Shen and Carolyn Rose. 2019. The Discourse of Online Content Moderation: Investigating Polarized User Responses to Changes in {R}eddit{'}s Quarantine Policy. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 58–69, Florence, Italy. Association for Computational Linguistics.

Carina Silberer, Vittorio Ferrari, and Mirella Lapata. 2017. Visually grounded meaning representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39.

K. Simonyan and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Samuel Smith and Quoc Le. 2017. A Bayesian perspective on generalization and stochastic gradient descent. *arXiv preprint arXiv:1710.06451*.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of ICLR (Conference Track)*.

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual nlp. In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015)*.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of ACL*.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of ACL*.

Kate Starbird and Leysia Palen. 2012. (how) will the revolution be retweeted?: information diffusion and the 2011 egyptian uprising. In *Proceedings of the acm 2012 conference on computer supported cooperative work*, pages 7–16. ACM.

Brandon M Stewart and Yuri M Zhukov. 2009. Use of force and civil–military relations in russia: an automated content analysis. *Small Wars & Insurgencies*, 20(2):319–343.

Ian Stewart, Yuval Pinter, and Jacob Eisenstein. 2018. Si o no, que penses? catalonian independence and linguistic identity on social media. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 136–141.

Philip J Stone, Dexter C Dunphy, and Marshall S Smith. 1966. The general inquirer: A computer approach to content analysis.

Reid Swanson, Brian Ecker, and Marilyn A. Walker. 2015. Argument Mining: Extracting Arguments from Online Dialogue. In *SIGDIAL Conference*.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 477–487. Association for Computational Linguistics.

Philip M. Taylor. 2003. Munitions of the Mind. *A history of propaganda from the ancient world.*

Oren Tsur, Dan Calacci, and David Lazer. 2015. A Frame of Mind: Using Statistical Models for Detection of Framing and Agenda Setting Campaigns. In *Proceedings of ACL-IJCNLP*, pages 1629–1638. Association for Computational Linguistics.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 248–258.

Andranik Tumasjan, Timm O Sprenger, Philipp G Sandner, and Isabell M Welpe. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Fourth international AAAI conference on weblogs and social media*.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 Task 3: Irony Detection in English Tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana. Association for Computational Linguistics.

Onur Varol, Emilio Ferrara, Christine L Ogan, Filippo Menczer, and Alessandro Flammini. 2014. Evolution of online user behavior during a social upheaval. In *Proceedings of the 2014 ACM conference on Web science*, pages 81–90. ACM.

Elham Vaziripour, Christophe Giraud-Carrier, and Daniel Zappala. 2016. Analyzing the political sentiment of tweets in farsi. In *Tenth International AAAI Conference on Web and Social Media*.

Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 2019. Do we really need fully unsupervised cross-lingual embeddings? *arXiv preprint arXiv:1909.01638*.

Ivan Vulić, Douwe Kiela, Stephen Clark, and Marie-Francine Moens. 2016. Multi-modal representations for improved bilingual lexicon learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 188–194.

Ivan Vulić and Anna Korhonen. 2016. On the role of seed lexicons in learning bilingual word embeddings. In *Proceedings of ACL*, pages 247–257.

Marilyn Walker, Pranav Anand, Rob Abbott, and Ricky Grant. 2012a. Stance Classification using Dialogic Properties of Persuasion. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 592–596, Montréal, Canada. Association for Computational Linguistics.

Marilyn A. Walker, Jean E. Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012b. A corpus for research on deliberation and debate. In *LREC*, pages 812–817. European Language Resources Association (ELRA).

Xiaojun Wan. 2008. Using bilingual knowledge and ensemble techniques for unsupervised chinese sentiment analysis. In *Proceedings of the conference on empirical methods in natural language processing*, pages 553–561. Association for Computational Linguistics.

William Yang Wang and William W Cohen. 2015. Joint information extraction and reasoning: A scalable statistical relational learning approach. 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)*, pages 355–364.

Zeerak Waseem, James Thorne, and Joachim Bingel. 2018. Bridging the gaps: Multi task learning for domain transfer of hate speech detection. In *Online Harassment*, pages 29–55. Springer.

Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, and Liqiang Wang. 2018. Improving the improved training of wasserstein gans: A consistency term and its dual effect. In *Proceedings of ICLR*.

Gregor Wiedemann. 2016. *Text mining for qualitative data analysis in the social sciences*, volume 1. Springer.

Roger D Wimmer and Joseph R Dominick. 2013. *Mass media research*. Cengage learning.

Fangzhao Wu and Yongfeng Huang. 2016. Sentiment Domain Adaptation with Multiple Sources. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 301–310, Berlin, Germany. Association for Computational Linguistics.

Chao Xing, Chao Liu, Dong Wang, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of NAACL-HLT*.

Ruochen Xu, Yiming Yang, Naoki Otani, and Yuexin Wu. 2018. Unsupervised cross-lingual transfer of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2465–2474. Association for Computational Linguistics.

Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. 2016. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11).

Pengcheng Yang, Fuli Luo, Shuangzhi Wu, Jingjing Xu, Dongdong Zhang, and Xu Sun. 2018. Learning unsupervised word mapping by maximizing mean discrepancy. *CoRR*, abs/1811.00275.

Yiming Yang, Thomas Pierce, and Jaime Carbonell. 1998. A study on retrospective and online event detection. In *Proc. ACM SIGIR*, pages 28–36.

Lori Young and Stuart Soroka. 2012. Affective News: The Automated Coding of Sentiment in Political Texts. *Political Communication*, 29(2):205–231.

Han Zhang. 2016. Physical exposures to political protests impact civic engagement: Evidence from 13 quasi-experiments with chinese social media. *Available at SSRN 2647222*.

Honglun Zhang, Liqiang Xiao, Yongkun Wang, and Yaohui Jin. 2017a. A generalized recurrent neural architecture for text classification with multi-task learning. *arXiv preprint arXiv:1707.02892*.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017b. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of ACL*.

Ayah Zirikly and Masato Hagiwara. 2015. Cross-lingual transfer of named entity recognizers without parallel corpora. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 390–396.

Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, Michal Lukasik, Kalina Bontcheva, Trevor Cohn, and Isabelle Augenstein. 2018. Discourse-aware rumour stance classification in social media using sequential classifiers. *Inf. Process. Manage.*, 54(2):273–290.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989.

Cornelia Züll and Peter Ph Mohler. 2001. Computerunterstützte inhaltsanalyse: Codierung und analyse von antworten auf offene fragen.