

THESIS

LINEAR MAPPINGS: SEMANTIC TRANSFER FROM TRANSFORMER MODELS FOR
COGNATE DETECTION AND COREFERENCE RESOLUTION

Submitted by

Abhijnan Nath

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Fall 2022

Master's Committee:

Advisor: Nikhil Krishnaswamy

Nathaniel Blanchard

Emily J. King

Copyright by Abhijnan Nath 2022

All Rights Reserved

ABSTRACT

LINEAR MAPPINGS: SEMANTIC TRANSFER FROM TRANSFORMER MODELS FOR COGNATE DETECTION AND COREFERENCE RESOLUTION

Embeddings or vector representations of language and their properties are useful for understanding how Natural Language Processing technology works. The usefulness of embeddings, however, depends on how contextualized or information-rich such embeddings are. In this work, I apply a novel affine (linear) mapping technique first established in the field of computer vision to embeddings generated from large Transformer-based language models. In particular, I study its use in two challenging linguistic tasks: cross-lingual cognate detection and cross-document coreference resolution. Cognate detection for two Low-Resource Languages (LRL), Assamese and Bengali, is framed as a binary classification problem using semantic (embedding-based), articulatory, and phonetic features. Linear maps for this task are extrinsically evaluated on the extent of transfer of semantic information between monolingual as well as multi-lingual models including those specialized for low-resourced Indian languages. For cross-document coreference resolution, whole-document contextual representations are generated for event and entity mentions from cross-document language models like CDLM and other BERT-variants and then linearly mapped to form coreferring clusters based on their cosine similarities. I evaluate my results on gold output based on established coreference metrics like BCUB and MUC.

My findings reveal that linearly transforming vectors from one model's embedding space to another carries certain semantic information with high fidelity thereby revealing the existence of a canonical embedding space and its geometric properties for language-models. Interestingly, even for a much more challenging task like coreference resolution, linear maps are able to transfer semantic information between "lighter" models or less contextual models and "larger" models with near-equivalent performance or even improved results in some cases.

ACKNOWLEDGEMENTS

I would like to thank my advisor Prof. Nikhil Krishnaswamy, a godsend to me and the coolest supervisor I have ever worked with. A brilliant computational linguist with a passion for his subject, his intellectual supervision and rigorous critique of my academic writing were priceless for my growth as a researcher. Without him, it would have been impossible to figure things out on my own. Who knew that one single guitar-salute back in Spring 2021 would turn into mutual clicks on an overleaf document a couple of semesters later?

I would also like to thank Prof. Nathaniel Blanchard and Prof. Emily King for their patience and overall supervision of my work. I am deeply grateful to Prof. Charles Anderson for his lessons on how artificial intelligence works at its core. Sincere thanks to luminaries like Prof. Martha Palmer and Prof. James Martin who offered crucial advice and insights into all-things computational linguistics through their regular RAMFIS meetings. I also offer my gratitude to Prof. Sanjay Rajopadhye for his exceptional insights into the computational aspects of a research problem and for his stories of growth, sacrifice and a fulfilling graduate school experience.

I am incredibly thankful to Rehan from CU Boulder whom I had the privilege of collaborating under the RAMFIS AIDA project. I learned a lot of things from him including clean coding, GitHub ethics, and having a cool mind. I am also very thankful to my friends at the Signal lab and Computer Vision lab at CSU for their never-ending support. Special thanks to Huma Jamil for always helping me out with little but important things like remote tunneling or docker-related frustrations, as well as for picking my brain with interesting questions about BERT stuff, Sonu for his advice on thesis writing, Jason for his insights on the math of language and of course, Mannan, Sina, and Ibrahim for their brainstorming sessions and support in my research. I am also thankful to Meredith, for all the things she taught me and her amazing company while I was writing research papers and my thesis.

DEDICATION

To my parents

Dr. Dambarudhar Nath and Dr. Jahnabi Gogoi

for always believing in me.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
DEDICATION	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
 Chapter 1	
INTRODUCTION	1
1.1 Language Models	1
1.2 Semantic Encoding and Transfer	4
1.3 Framing the Problem	4
 Chapter 2	
RELATED WORKS	6
2.1 Cognate Detection	6
2.2 Cross Document Coreference Resolution (CDCR)	7
2.2.1 Entities	8
2.2.2 Events	9
2.3 Equivalence of Embedding Spaces	11
 Chapter 3	
COGNATE DETECTION	13
3.1 Background	13
3.2 Assamese-Bengali Cognate Datasets	15
3.2.1 Designing the Wiktionary Scraper for gold standards	15
3.3 Semantic Similarity	18
3.3.1 The Cosine Function	18
3.3.2 Selection of Transformer-based BERT Variants	19
3.3.3 AxomiyaBERTA	21
3.3.4 Word and Sentence Embeddings	22
3.3.5 Affine Transformations Between Embedding Spaces	24
3.3.6 Semantic Feature Exploration: A Sanity Check!	25
3.4 Non-semantic features	30
3.4.1 Articulatory features: Alignment Network	31
3.4.2 Orthographic and Phonetic Edit Distances	31
3.5 Evaluation and Analysis	33
3.5.1 Classifier Models and Feature Abbreviations	33
3.5.2 Influence of Features	33
 Chapter 4	
CROSS DOCUMENT COREFERENCE RESOLUTION (CDCR)	42
4.1 Background	42
4.2 LDC Phase 1 Dataset and Knowledge Graphs	43
4.3 CDLM Model and other variant selection	47
4.4 Bi-Encoder for Document-level embeddings	51

4.5	Generating Affine Maps	55
4.6	Pairwise Clustering and Thresholding	56
4.7	Coreference Evaluation Metrics	57
4.8	Results and Analysis	58
Chapter 5	CONCLUSION AND FUTURE WORK	65
Bibliography		68

LIST OF TABLES

3.1	Assamese-Bengali sound correspondences.	14
3.2	Cognate pair counts per language.	15
3.3	Number of Hard-Negatives (HN), Synonyms (Syn.), Cognates (Cog.), and Random pairs (Rnd.) in Assamese-Bengali and Bengali-Assamese train/test sets.	18
3.4	AxomiyaBERTA Model configuration trained on a monolingual Assamese corpus. . .	22
3.5	Abbreviations for feature combinations.	34
3.6	NN classifier results (as %) for the ed-dl-ab-am feature combination (full feature set).	35
3.7	Sample false negatives.	35
3.8	F1(+) as % with and without alignment score (dl) and Levenshtein distance features for NN model	35
3.9	Number of false positives using ed vs. ped-b feature sets broken down by negative example type (hard negative, synonym, random). Bidirectional and pair-specific models shown.	38
3.10	Number of false positives in pair-specific model outputs using ped, ped-m (pm), and ped-m-sm (psm) feature sets broken down by negative example type (hard negative, synonym, random).	40
3.11	Number of false negatives (undetected cognates) in pair-specific model outputs using ped, ped-m (pm) and ped-m-sm (psm) feature sets.	40
4.1	Table showing distribution of various features of the LDC dataset for events and entities. Note that the label "Unique lexical" represents lexically unique mentions. For instance, if "murdered" as an event mention occurs twice in the entire dataset and they do not corefer, then their respective gold cluster IDs will be different. Also, in that case, "murdered" will only be counted once in the "Unique lexicals" feature. Similarly, if a gold cluster occurs twice, it is only counted once in the Unique gold clusters label.	44
4.2	Final BCUB and MUC Scores i.e., Precision (P), Recall (R) and F1-scores (F1) for native and mapped embeddings on LDC events . CDLM, BERT_BASE (B_B), BERT_LARGE (B_L), BERT_COREF (B_C) are native models. Their affine-mapped variants are shown as Source→Target in the Model Variants column.	59
4.3	Final BCUB and MUC Scores i.e., Precision(P), Recall(R) and F1-scores(F1) for native and mapped embeddings on LDC entities . CDLM, BERT_BASE (B_B), BERT_LARGE (B_L), BERT_COREF (B_C) are native models. Their affine-mapped variants are shown as Source→Target in the Model Variants column.	61

LIST OF FIGURES

3.1	Sample equivalent sentences with cognate words (and English translations) underlined.	23
3.2	Cross-embedding space mapping pipeline resulting in directly comparable vector representations (MBERT→AxomiyaBERTA used as example).	25
3.3	Kernel Density Estimation Plot for semantic similarity between word pairs generated as native embeddings from each of the four models, colored by pair type.	27
3.4	Pairwise Plots for semantic similarity between word pairs generated as native embeddings from each of the four models, colored by pair type.	28
3.5	Kernel Density Estimation Plot for affine-mapped Word and Sentence Embeddings. Note: AxomiyaBERTA has been shown as AXBerta to fit the subplot legend.	29
3.6	Pairwise Plot for affine-mapped Word and Sentence Embeddings. Note: AxomiyaBERTA has been shown as AXBerta to fit the subplot legend.	30
3.7	Entire Pipeline for Cognate Detection Task with Phonetic, Articulatory and Semantic Features Note that the Alignment Network and the 3 Layer DNN are two separate neural networks, the former being used for getting alignment scores while the latter for classification.	32
3.8	Influence of different semantic feature sets compared to phonetic edit distance baseline (ped).	36
4.1	Example showing coreferring and non-coreferring links for events in LDC Phase 1 Dataset.	43
4.2	Example showing a Mention Map (events) knowledge graph in LDC Phase 1 Dataset	47
4.3	Global and Local attention that the CDLM model uses. [Beltagy et al., 2020]	50
4.4	Entire CDCR Pipeline for Entities with details of the linear mapping technique in the segment below. Same pipeline applies for event mentions.	51
4.5	Cross-document Event/Entity Coref Algorithm.	53
4.6	Principal Component Analysis (PCA) of Event embeddings generated from the default CDLM model. The left figure is a snapshot of the 3-D PCA while the right figure is a 2-D PCA snapshot. Red markers point to the visible event clusters.	54
4.7	Principal Component Analysis of Entity embeddings generated from the default CDLM model. The left figure is a snapshot of the 3-D PCA while the right figure is a 2-D PCA snapshot.	55

Chapter 1

INTRODUCTION

1.1 Language Models

Language is a *temporal* phenomenon that is linguistic meaning (semantics) is a consequence of an ordered arrangement of words. A sequence of words in a sentence "makes sense" because of where they are placed and their relative positions in the sequence. In other words, their positional characteristics are directly related to the semantic meaning and grammatical correctness of a sentence. Therefore, in Natural Language Processing (NLP), a language model is a probabilistic model that uses probabilistic constraints and assumptions to determine the probability of a word given its surrounding words. For instance, early language models like the N-gram [Jurafsky, 2000] use the "memoryless" Markov assumption [Markov, 1954] to approximate the probability of a sample word given its history or surrounding words.

However, N-gram models are based on words and are very literal. Such models robustness is also heavily dependent, or tend to be overfitted [Everitt and Skrondal, 2010], on the training corpus. In other words, N-gram probability matrices, often represented as one-hot encodings¹, are incredibly sparse. Take a vocabulary V with N samples or words. The total combinations of unique bigrams (two-word sequence of words) alone are N -squared without considering any higher order n-grams like tri-grams. This makes it difficult for such models to learn coherent sentences or "context" unless the vocabulary is huge which is not always the case depending on the task. Apart from this sparsity, they also have a *closed* vocabulary meaning they have difficulty approximating probabilities for unknown or Out-Of-Vocabulary (OOV) words especially at inference time. The sparsity problem and the concomitant curse of dimensionality [Verleysen and François, 2005] as well as the limited vocabulary related problems were mitigated to some extent by the advent of neu-

¹One-hot encodings are sparse and do not preserve similarity across dimensions thereby making them limited in their information content.

ral networks. This is because words were now represented in a continuous parameter space in the form of mathematical vectors or embeddings.² Neural frameworks provided enough "learnable" parameters to not only map a majority of the unique combinations of words in the vocabulary but also mitigated the problem of unknown (OOV) words by making such unknown word-embeddings "learnable". Moreover, a sliding window approach to learning word probabilities offered more flexibility in "context" for neural networks than simple N-gram models.

There was still an underlying problem. As it turns out, there is a crucial issue in these methods: the case of a limited context, before as well as after the word being learned as an embedding. Language being inherently temporal requires an artificially-intelligent system to have a greater context for it to learn efficiently. This problem was relatively alleviated by Recurrent Neural Networks (RNNs) and more so with the Long-Short Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997]. These networks, in general, saw architectural inventions like *memory gates* which allowed selective sequential processing of linguistic-information. For instance, LSTMs typically had a forget gate that let only relevant information to be learned while ignoring superfluous information. In other words, they could now peek into a relatively long history of prior words in a text to utilize the temporality of language. However, the solution itself (its architecture) became a problem even though increasing RNN layers should have ideally let us look even further behind in the text. Widespread machine learning (ML) hacks such as stacking RNN layers for chasing the state-of-the-art also led to problems like the vanishing gradient³ with significant loss of information. Moreover, since each step needed recurrent processing and simultaneous processing of steps was not possible, parallelizing the training process was difficult [Jurafsky and Martin, 2009].

There was another problem with RNNs. That is, they typically allowed a *one-directional* sequence processing of language. That is, a word coming after has no effect on the word being

²Although historically borrowed from Information Retrieval theory, an embedding in NLP is usually a real-valued representation of a word for text analysis. In most cases, it encodes word-semantics or meanings. As such, words that are closer in the vector space are expected to be similar in meaning. Note that sentences or whole-documents can also be expressed as embeddings.

³Imagine multiplying a number ($\ll 1$) a hundred times. You effectively end up with a very small number almost close to *zero* which nullifies the gradient-flow during back propagation of weights. LSTMs were relatively immune to this problem than simple RNNs, but not entirely even if they were designed to solve this problem.

processed. This was non-intuitive from a human understanding or "semantic" perspective since we usually understand the meaning of a text by reading the entire text in context. The importance of word-semantics to capture the "context" of texts led to the emergence of the classic NLP word-embeddings like Word2Vec [Mikolov et al., 2013], Glove [Pennington et al., 2014]. These neural models represent words as real-valued, fixed dimensional dense vectors while preserving similarity between similar words. They were crucial for NLP advances in the early part of the last decade especially for tasks like synonym detection, finding analogies etc., that required semantic information. The catch was, they still could not tackle semantic polysemy. In other words, these static⁴ embeddings typically cannot differentiate between similar-looking words that mean different things in different contexts. For instance, "bank" as a financial institution vs "bank" of the river.

The advent of Transformers [Vaswani et al., 2017] brought forth the idea of using "contextualized" embeddings for NLP tasks instead of the static variants. Transformers are unique in that they do not rely on recurrent connections for sequence processing but they are somewhat similar to feedforward networks, only more layered. A crucial tool in their arsenal, the "self-attention mechanism", is a clever way to extract and use information from arbitrarily large contexts without the need to feed it through recurrent connections. In other words, computations in a sequence for a word are now *independent* of similar computations for the rest of the words. Without getting into the technical details, this allows the network to learn from the "context" of a word in a sequence in a parallelized fashion. The latter characteristic was quite conducive for training with Graphics Processing Units. State-of-the-art language models like the BERT [Devlin et al., 2018] use Transformer encoders to learn a contextualized representation of language. BERT variants come in all shapes and sizes with even task-specific variants. However, one common trend of exploration is whether and to what extent semantic information is encoded in such models embedding space.

⁴These models learn a single vector embedding for each unique word w in the vocabulary without considering the semantic-context of the word.

In this thesis, I explore to what extent semantic information is captured in large language models (LLMs), including those focused on certain specific Indian languages as well as task-specific or "more contextualized" variants. More specifically, I explore whether such semantic information can be transferred by an *affine-mapping* procedure between native and derived (affine-mapped) embedding spaces. I evaluate using an *extrinsic* [Jurafsky, 2000] method of evaluation on two downstream tasks: cognate detection and cross document coreference resolution.

1.2 Semantic Encoding and Transfer

The research question about Transformer-based language models learning the "semantics" or "meaning" of a text has become very popular [Tenney et al., 2019], [Jawahar et al., 2019a]. It has also been approached from a geometric angle with vector-space related explorations in works such as [Ethayarajh, 2019a]. Assuming large language models do capture task-based semantics, my thesis addresses the question of how geometric techniques can be used to transfer semantic information between language models with a high fidelity. In other words, can semantic information be transferred between models using a linear transformation and if so, what does this tell us about the geometric properties of the embedding spaces of the native and mapped models?

1.3 Framing the Problem

We chose an affine (linear) mapping technique between embeddings from our selected models because linear transformations are *homomorphic* i.e., they preserve spatial structure between the points that represent the embedding. The idea is that if two embedding spaces capture semantic information equivalently, a "structure-preserving" unsupervised linear mapping⁵ should be

⁵Non-linear maps could still encode similar features from both spaces as a consequence of correlation but they will not share a representation i.e., non-linear transformations which are not homomorphic or structure preserving, will naturally perturb the chances of embedding vectors encoding semantic information in a *distinct* space [McNeely-White, 2020]. Also, introducing non-linear maps in case of language model embeddings is trickier than image-based CNN embeddings since such models are huge in their parameter space (>350M for BERT-LARGE vs. 25.6M for ResNet-50) with resulting GPU-related complexities. Even though our model space includes smaller monolingual models where a non-linear mapping experiment might sound more feasible, our aim is focused more on how much semantic information is robustly transferred while keeping the *linearity* of maps consistent.

able to transfer such information between embeddings notwithstanding the non-linear fashion in which the embeddings were originally learned in the language model. If V_s^i and V_t^j represent the **source** and the **target** embedding spaces, we carry out a least-square-based optimization method or ridge-regression [Hoerl and Kennard, 1970] between those spaces where i and j represent individual samples in our task-specific datasets while assuming $\lambda=1$. Here, β represents the L^2 -norm regularization parameter. Mathematically,

$$\text{Minimize}((V_t^j - V_s^i\beta)^T(V_t^j - V_s^i\beta) + \lambda\beta^T\beta)$$

At inference time, we ensure consistency by evaluating the affine maps between the source and the target space using the same classifier (cognate task) or clustering method (CDCR) for each of the mappings. This ensures that no outside-information or information-mixing occurs during our experiments. Our hypothesis is that, similar or equivalent performance using native or mapped embedding spaces along with the consistency of using the same inference tools (same parameters) should help us verify the nature and extent of semantic information transfer between the embedding spaces. Also, since the language models we have selected differ in their architecture, training corpora, hyperparameters as well as their objective (loss) functions, such semantic transfer will help us understand the nature of the distinct semantic space within which such information lies. It will also reveal insights into whether such semantic space is task-specific or agnostic and how dispersed they are in the high-dimensional vector space [Ethayarajh, 2019a]. This is because the selected language models were trained for a more generalized task of mask-word prediction [Devlin et al., 2018] while our extrinsic evaluation is conducted on two tasks that are relatively *different* from the masked word prediction based training.⁶

⁶Prior works such as [McNeely-White et al., 2022], [McNeely-White, 2020] that have inspired this work studied image or face embeddings from models that were trained as well as *evaluated* (with affine maps) on the same task that the chosen models were trained on i.e., their methodology was relatively more supervised than this language-model based work. This point of difference is important as it has consequences on the generalizability and task-agnosticity of language representation models.

Chapter 2

RELATED WORKS

In this section I briefly discuss the previous works in the field of i) cognate detection, ii) cross document coreference resolution for event and entities, iii) works associated with the affine mapping technique for linguistic as well as image-based embeddings as well as how our work adds in to the recent developments.

2.1 Cognate Detection

There have been many different approaches for detecting cognates within the NLP community. In most previous works, researchers have utilized some form of string-distance or similarity metric based on phonetic, orthographic, language-specific sound changes or semantic or a combination of all to detect cognates. For instance, a major early work by [Kondrak, 2001] identifies cognates in Algonquian using phonetic and semantic similarity, the latter using lexical resources like WordNet [Miller, 1995]. [Mulloni and Pekar, 2006] infer orthographic changes resulting from inheritance of words in languages and uses an orthographic distance metric to detect cognates. This is a *step-up* since this circumvented the need for lexical resources like bi-lingual dictionaries.

Later works such as [Jäger, 2018] evaluates Pointwise Mutual Information (PMI) and Support Vector Machine (SVM)-based methods in cognate detection over the Automated Similarity Judgment Project (ASJP) database [Brown et al., 2008]. [List, 2014] finds relationships between factors like data size and genetic relatedness in automated cognate detection between English, German, Dutch, and French while also creating a gold-standard cognate dataset in those languages. [Bloodgood and Strauss, 2017] explore using global constraints and re-ranking of score matrices to improve such detection. Works such as [Dellert, 2018] explores sequence alignment and phonetic correspondence features for Northern European languages, two feature types that are very relevant to our work. Slightly different but related works such as [Rama et al., 2018] and [Rama and List, 2019] turn the problem around and explore the direct applicability of automated cognate detection

methods to phylogenetic reconstruction and inference. Lexical resources like WordNet make a comeback in this field in [Kanojia et al., 2021a] where they perform orthographic similarity-based cognate detection in various Indian languages, but notably not Assamese (one of the languages I investigate in this thesis).

More related to the articulatory features that we use for our experiments are works like [Bharadwaj et al., 2016] and [Rijhwani et al., 2019] that suggest that phonologically-aware articulatory representations from PanPhon [Mortensen et al., 2016] can either be used natively as embeddings or as features in attention-based neural models for downstream NLP tasks such as Named Entity Recognition (NER) or entity linking for low-resource languages. [Labat and Lefever, 2019] and [Lefever et al., 2020] add semantic information to orthographic features for cognate detection to get substantial improvement in resource-rich languages like English and Dutch (90% F1). Similarly, [Kanojia et al., 2021b] explores mixing large multilingual model embeddings to cognitive features like gaze improves cognate detection in low-resource languages like Hindi and Marathi (86% F1). For the hybrid approach to similarity metrics, work in translation lexicons (e.g., [Schafer and Yarowsky, 2002]) offers newer perspectives to cognate detection related features.

For my thesis, I primarily explore the contributions of semantically-mapped features while touching on the non-semantic features i.e., phonetic, orthographic, and articulatory alignment to the task of cognate detection between Assamese and Bengali (low-resourced Indian languages). I use a comparative method topped with ablation-based experiments.

2.2 Cross Document Coreference Resolution (CDCR)

Although the task of coreference resolution has been historically studied for resolving both events and entities, their respective trajectories of evolution sometimes align and at times diverge due to either the 1) the differing difficulty of one from the other (event coreference resolution being more difficult [Humphreys et al., 1997]), 2) the availability of annotated datasets for both or individually, 3) the differing availability of lexical, semantic as well as neural resources or

pretrained models for entities and events. In this section, we provide a rough overview of the related works in both event coreference as well as entities while also including works that cater to their joint resolution. Moreover, in a purely chronological sense, within-document (WDCR) coreference (considered to be easier than cross-document [Zhukova et al., 2021]) has been actively tackled by the community earlier than the cross document (CDCR) variety because the latter is not only computationally harder due to a larger search space but also conceptually more tricky as such task-based models require an understanding of multiple articles and their dependencies.

2.2.1 Entities

Early works such as [Soon et al., 2001], [Ng and Cardie, 2002], [Rahman and Ng, 2011] applied machine learning approaches to coreference resolution but they mostly focused on hand-engineered features such as relying on noun phrases and pronouns while not catering to events or entities in general. Later works such as [Denis and Baldridge, 2007], [Durrett and Klein, 2013] and [Björkelund and Kuhn, 2014] used some form of *neural* mention-ranking models or cluster-ranking models. For instance, [Durrett and Klein, 2013] suggested lessening the use of *heuristics* (hand-engineered features) for capturing syntactic, discourse-related and semantic properties of language. Instead, they extended their feature set to include shallow linguistic properties of coreferring mentions like head word, string-lemma matching as well as spatial properties like sentence-distance between mentions. However, even if their model could do well when capturing syntactics and discourse, their model does not capture linguistic semantics very well. Similarly, [Björkelund and Kuhn, 2014] experimented with approximate beam search algorithms on perceptron layers that take in "non-local" features such as the linear shape of the mention, mention-syntactics (i.e., whether they appear as objects or subjects) as well as the *recency* [Jurafsky, 2000] of mentions. They also include previously popular syntactic and lexical features which they call as "local". Their evaluation on three languages i.e., Arabic, Chinese and English led them to achieve significant improvement over their baselines on CoNLL 2012 Shared Task.

With the advent of using *semantic* information from word-embeddings brought forth by [Mikolov et al., 2013], entity-based coreference resolution work [Clark and Manning, 2016] suggested a neural clustering system. This was based on a distributed representation of mentions that involves word embeddings, additional positional features of mentions, document genre, and speaker-based features. Their model further outperformed the prevailing state-of-the-art on the English and Chinese portions of the CoNLL 2012 Shared Task dataset.

2.2.2 Events

When it comes to event coreference resolution things start to get a little more complicated. Resolving events is more difficult than entities due to the complexities involved in the appearance and structure of event mentions. Technically, entities are mostly noun phrases while events may consist of a verbal predicate ("acquire") or a nominalization ("acquisition") while simultaneously being attached to arguments such as event participants and spatio-temporal information, and more [Barhom et al., 2019]. Moreover, event coreference is more complex because it is secondary to entity (or object) coreference. That is, two events are usually coreferential due to two entities being coreferential for that actual event pair [Humphreys et al., 1997].

Early works such as [Humphreys et al., 1997], [Bagga and Baldwin, 1999], [Chen and Ji, 2009] in event coreference used supervised training based on lexical and linguistic features. For instance, the parts-of-speech tags, phrasal-matching or structural features like aligned arguments. These works compute some form of a reliable distance between the event mentions based on vector space models including VSM-Disambiguate module [Salton, 1989] and thereby used different forms of graph-clustering techniques to form coreference cluster-links. Perhaps one of the first papers to approach joint neural modeling of both events and entities in a cross-document setting, [Lee et al., 2012] uses linear regression to control the flow of information between entity and event clusters. This was based on the hypothesis that resolving coreference for events should benefit from information coming from their arguments (or entities) and vice versa. They were able to improve previous baselines by over 3 CoNLL F1 points.

Later works such as [Bejan and Harabagiu, 2014], [Yang et al., 2015] explored various forms of nonparametric Bayesian models using lexical features such as head-word and their lemmas, their context, class-based linguistic features such as parts-of-speech (POS), word-class features based on TimeML specification language [Pustejovsky et al., 2003], WordNet-based features, as well as semantic features from PropBank [Palmer et al., 2005], Word2Vec [Mikolov et al., 2013] etc. [Kenyon-Dean et al., 2018] used neural models trained on *customized* objective (loss) functions to generate richer representations of mention-pairs using features such as contextual Word2Vec embeddings, document level features such as TF-IDF as well as comparative features between mentions like mention-recency, word overlap, and lemma overlap etc. Their primary contribution was the new objective function called CORE which essentially punishes the model for differing coreference clusters of mentions (events, not entities) and awards it for similar clusters using cosine distance as the function. They improved upon the the results of Cybulska and Vossen (2015) on ECB+ corpus. Another interesting piece of work that caters to both events and entities following [Lee et al., 2012] is [Barhom et al., 2019] where they experimented with joint-modeling of events and entities by jointly-clustering events mentions with their related entity clusters by using predicate-argument structure. Their disjoint model improved the previous state of the art of [Kenyon-Dean et al., 2018] by 9.5 F1 points using the CoNLL scorer whereas their joint model improved upon the disjoint model by 1.2 points for entities and 1 point for events.

Most recent works in cross-document event coreference like [Yu et al., 2022], [Caciularu et al., 2021], [Cattan et al., 2021a] and [Zeng et al., 2020] suggest use of *representation learning* models. These methods use distributed and contextually-enriched vector representations of mentions for both within a sentence or the whole document. Such representations are usually from large language models like various BERT-variants to calculate supervised pairwise scores for those event mentions. Thereafter, they use some form of incremental or agglomerative clustering techniques to form predicted coreference links and evaluate their chains based on popular coreference metrics like BCUB, MUC scores. More specific to our research is the CDLM model [Caciularu et al., 2021], which uses a combination of enhanced pretraining using the *global* attention mechanism

inspired by [Beltagy et al., 2020] as well as finetuning on a task-specific dataset using pretrained special tokens to generate more semantically-enhanced embeddings for mentions. While it has been useful for downstream NLP task like CDCR, question-answering etc, we intend to compare how our linear mapping technique fares for CDCR when using embeddings from the CDLM model and three BERT variants. Task-specific details for CDCR are in chapter dedicated to CDCR.

2.3 Equivalence of Embedding Spaces

Most prior works to compare and contrast neural networks in the domain of computer vision have applied a black box method of evaluation where various CNNs are sampled and evaluated against common benchmark datasets like ILSVRC2012⁷. While this comparative method is very popular in domains like NLP as well, one of its disadvantages is that not only does it, at times, lead to a blind chase for the state of the art [Church and Kordoni, 2022] by focusing on little incremental gains in performance but also fails to provide a deeper comparison of models when their performance is very similar [McNeely-White et al., 2020a].

Previous works that use mapping between embedding spaces such as [Lenc and Vedaldi, 2015] suggest that affine transformation between intermediate-layer embedding spaces of popular CNNs expose properties like *equivalence* and *equivariance*. However, their method is supervised since they use labels in training their affine transformations thereby creating newly trained networks and the concomitant increase in the observation and evaluation space for parameters.

Later works in the field of computer vision take it further by carrying out unsupervised affine mappings between CNN models penultimate layer embedding. For instance, [McNeely-White et al., 2020a] suggests that there are canonical and task-based similarities between the features extracted by two convolutional neural network (CNN)-based models like Inception [Szegedy et al., 2015] and ResNet [He et al., 2016] *despite* the difference in architectures and hyperparameters of the two models. They provide empirical evidence that affine (linear) transformation across embedding spaces of these two models transfer similar task-specific information which is proven by the

⁷<https://www.image-net.org/challenges/LSVRC/2012/>

minimal performance penalty across image classification tasks. Similar works by [McNeely-White et al., 2022, McNeely-White et al., 2020b] further buttress their results that CNN feature spaces are interchangeable since they discover similar task-specific features and can be exposed by linear transformations. They also extend their equivalence of CNN embeddings spaces to face embeddings and suggest critical security-centric implications if such equivalence holds true. However, they mostly extract features from the penultimate, non-classification layers of their chosen CNNs and while their results are promising, their method does not explore if such equivalence exists between earlier or intermediate layers of CNNs.

Not much work has been done in embedding space equivalence or semantic transfer for language models especially those for low resource languages. Works such as [Ganesan et al., 2021] and [Artetxe et al., 2018a, Artetxe et al., 2018b] improve bilingual lexical induction using either linear or non-linear word embedding maps, but they use non-contextual "static" embeddings like fastText⁸ or Word2Vec. In my thesis, I extend the previous works done in the computer vision domain such as [McNeely-White, 2020, McNeely-White et al., 2022] to language models to study semantic encoding and transfer using contextual word embeddings from Transformers, unlike the static embedding work done previously.

⁸<https://fasttext.cc/>

Chapter 3

COGNATE DETECTION

The contents of this chapter were presented in [Nath et al., 2022] at the Ninth International Workshop on NLP for Similar Languages, Varieties, and Dialects (Vardial 2022, ACL). In this thesis, I extend the presented work to emphasize the relative contribution of semantic features compared to non-semantic features for cognate detection while also exploring the semantic encoding and transfer property between the embedding spaces of native and affine-mapped versions.

3.1 Background

Linguistically, cognates are words that are inherited by direct descent from a common etymological ancestor in a common parent language [Crystal, 2011]. The parent Latin term "*cognatus*" that they derive from means "blood relative" [Ammer, 2013]. Within a bi-lingual framework, a cognate pair is a pair of words in the individual languages that are both inherited from the same word in the same parent language.

Naturally, some quintessential qualities of cognates are their semantic affinity, phonetic affinity or pronunciation overlaps. However, they do not necessarily need to carry the same meaning because of historical sound changes and usage in a specific language. For instance, English "starve" and Dutch "*sterven*" ("to die") all derive from the same Proto-Germanic root, **sterbaną* ("die")⁹. Interestingly, they do not always need to have similar phonetics either. For example, English "two" is cognate with Armenian *erku*, as both are descended from Proto-Indo-European **dwóh₁*, with **dw->>tw-* and **dw->>erk-* being regular, if non-intuitive, parallel sound changes. Also, since cognates are inherited and not borrowed from a separate language like loanwords are, they are necessarily subject to diachronic sound change.

⁹<https://en.wikipedia.org/wiki/Cognate>

Due to the above characteristics of cognates including phonetic and semantic shifts, detecting cognates is quite challenging. A necessary first step could be to distinguish cognates from other classes of words like ordinary translations or words that share a trivial phonetic affinity. However, if such detection is possible, it can be used to discern the evolutionary paths of related languages thereby making them very useful for historical linguists. In this part (cognate-detection task) of my thesis, I include as well as extend the work from my paper [Nath et al., 2022] while focusing more on the semantic encoding and transfer related aspects.

Our task is cognate detection between two closely-related languages: Bengali (ISO code `bn`) and Assamese (ISO code `as`). Bengali (262 million speakers) and Assamese (15 million speakers) are two major languages of eastern India and Bangladesh. From a scriptural perspective, they share a common descent from Early Indo-Aryan via Magadhi Prakrit, both being written using Bengali or Eastern Nagari script. The reason for choosing these two languages is two-fold: firstly, they are both low-resource languages (LRL), Assamese a tad more than Bengali. Secondly, even though this language pair has a lot of scripture-based and phonetic similarities, the challenge is to have a classification system that is good enough to overcome the important differences in their sound patterns. For example, both the languages share certain grammatical features like classifying affixes (e.g., Asm. `-zon`, Beng. `-dʒon`, referring to persons), as well as certain common phonetic innovations (such as the evolution of Sanskrit $/ə/ \rightarrow /ɔ/$). Similarly, there are differences primarily in the pronunciation of consonants. For instance, $/s/$ in Sanskrit is lenited to $/x/$ in Assamese whereas Bengali palatalized it to $/ʃ/$. Table 3.1 shows a few examples of such transformations.

Assamese	Bengali
s,s,z,z	tʃ,tʃ ^h ,dʒ,dʒ ^{fi}
t,t ^h ,d,d ^{fi}	t̪/t̪ ^h /t̪ ^h ,d̪/d̪ ^{fi} /d̪ ^{fi}
x,ɪ	ʃ,r

Table 3.1: Assamese-Bengali sound correspondences.

3.2 Assamese-Bengali Cognate Datasets

3.2.1 Designing the Wiktionary Scraper for gold standards

For our purpose of constructing a gold-standard cognate dataset, we stick to the linguistic definition of cognates that treats cognates and loanwords (borrowed words) as distinct and not subsumed (e.g., [Kondrak, 2001]). Even if there might be overlaps i.e., our cognate gold standard might contain samples that are in reality loanwords, this is effectively negligible to the best of our knowledge. Cognates in Bengali and Assamese must share a common descent from an ancestor language and the best-documented is Sanskrit.¹⁰ However, many descendants of Sanskrit that could be potential cognates make scholarly reborrowings from classical Sanskrit (*tatsama*). These are fully reincorporated Sanskrit forms adapted to fit the modern phonology. On the other hand, *tadbhava* words, that exist alongside with the former, are inherited from Old Indo-Aryan with concomitant sound changes in the Middle Indo-Aryan phase.

Wiktionary turned out to be accessible and well-resourced for this data collection process. Categories of the form [Descendant]_terms_derived_from_Sanskrit were scraped for each of the two descendants.¹¹ Thereafter, we did a union operation of these two sets and then took the subset¹² of the union where both the Assamese and Bengali forms had the same documented Sanskrit ancestor. Table 3.2 shows the number of cognates retrieved for each language.

Descendant	Ancestor	# Cognates
Assamese	Sanskrit	205
Bengali	Sanskrit	335

Table 3.2: Cognate pair counts per language.

¹⁰The transitory vernacular "Aṣṭadhyāyī" could have been an alternative parent language for our task since it spans the period between the 6th and 13th centuries CE. However, annotated resources for this are hard to come by on the web.

¹¹e.g., https://en.wiktionary.org/wiki/Category:Assamese_terms_derived_from_Sanskrit

¹²Note that despite the union-intersection operations being symmetrical, the dataset sizes differ for the two languages by a small number of entries; because the count of overall entries for Bengali are comparatively higher in the English Wiktionary, there are more cases where multiple Bengali words have the same ancestor as a single documented Assamese word.

Every word in every pair was then converted to its phonetic representation in the International Phonetic Alphabet (IPA) using the EpiTran package [Mortensen et al., 2018]. The IPA is a standardized representation of phonetic notation written primarily in Latin script. In NLP and speech processing, such standardized representations for lexical, as well as prosodic elements of sound, help in extracting language or speaker-specific distinctiveness in the feature engineering process. The available EpiTran distribution does not support certain low-resourced languages, among them Assamese, but the format is easily extensible. We wrote an EpiTran graph-to-phoneme mapping for Assamese using resources like Omniglot¹³ and Wikiwand/Assamese¹⁴, as well as native speaker guidance for verification.

In order to make the process of evaluating our classifiers more rigorous we also gather word pairs that are not cognates including adversarial samples such as phonetic neighbors. More specifically, these are: i) **hard negatives**: phonetically similar non-cognates; ii) **synonyms**: semantically similar words, like ordinary non-cognate translations; iii) **randoms**: pairs where the two words have no discernible phonetic or semantic relationship.

In order to source **hard negative** examples, the PanPhon package [Mortensen et al., 2016] was leveraged to calculate six different edit distances between the IPA transcription of every gathered cognate in one language, and the IPA transcription for every lemma in the other language (the list of lemmas was also scraped from Wiktionary). Lemmas help leverage the phonetic similarity between the actual cognate word and its reduced inflectional forms to create phonetically similar but non-cognate adversarial examples. For each edit distance metric, we select the word that has the lowest edit distance to the cognate in question. This results in up to six hard negatives per cognate. However, it could be less if more than one edit distance metric returns the same nearest neighbor. Example IPA forms of hard-negatives in the respective languages: Asm. কথা (/kɔtʰa/) “word”, Beng. কতো (/kɔtʰa/) “how many”.

¹³<https://omniglot.com/writing/assamese.htm>

¹⁴<https://www.wikiwand.com/en/Help:IPA/Assamese>

Synonyms or words with similar meaning can also act as adversarial samples in cognate detection. As such, we scraped Wiktionary metadata and retrieved synonyms for each word in the collected cognates list where available. Example IPA representations of synonyms in each language: Asm. কুটুম (/kutum/) “family”, Beng. ব্ৰিহতদাৰ (/riʃtadar/) “relatives.”

Since any decent classifier needs to disambiguate random correlation between samples in their decision boundaries, we created the **randoms** pairings by pairing each cognate with a random word in the other language. We also ensured that any intersections among these three negative samples and between these and the cognates dataset were eliminated as and when it occurred.

In order to create sentence-sensitive embeddings for testing semantic transfer between source and target model spaces, we also manually create simple sentences including the cognate words in our dataset. In doing this, we ensured parts of speech consistency and correctness while also keeping the meaning of the cognate word consistent with the sentence. Including a native speaker verification process helped us in this task. (See Sec. 3.3.4 for more details.)

These subsets were then joined to create two language pair-specific dataset and one bidirectional dataset. 1) Assamese-Bengali¹⁵, where the Assamese word is the baseline comparand to which the Bengali word is compared. 2) Bengali-Assamese, where we reverse the case.¹⁶ 3) All-languages. This one is a simple concatenation of the above two and bidirectional. The idea is to allow the final classifier to learn from similarity metrics that flow in both directions.

Table 3.3 gives breakdown of the Assamese-Bengali and Bengali-Assamese Train and Test splits based on their labels. As we have used the linguistic definition of cognates as being distinct from loanwords, the latter may exist in the other categories. Since loanwords are usually phonetically similar to their sources, where loanwords do exist in our data, they are overwhelmingly likely to be in the hard negative category.

¹⁵For ablation tests, we also create subsets from a training and inference perspective for Assamese-Bengali and Bengali-Assamese which are herein denoted in tables and charts with an asterisk (*) or additional label `train_ev`.

¹⁶Since the order of words are flipped here, the edit distances are naturally symmetric but our alignment scoring is done for randomized splits of the data and attends to *directional* articulatory flow. As such, alignment scores will differ upon the reversal of word pairs.

	<i>as-bn</i>		<i>bn-as</i>	
	train	test	train	test
Cog.	306	303	306	300
HN	776	769	721	716
Syn.	329	327	317	316
Rnd.	304	301	304	299
Total	1715	1700	1648	1631

Table 3.3: Number of Hard-Negatives (HN), Synonyms (Syn.), Cognates (Cog.), and Random pairs (Rnd.) in Assamese-Bengali and Bengali-Assamese train/test sets.

3.3 Semantic Similarity

Calculating semantic similarity for word-embeddings has been approached from many angles in the literature. However, one of the most popular methods to compute such semantic similarity or semantic distance has been the cosine function¹⁷ between two vectors. Since the dawn of word-embeddings as distributed representations in low dimensional continuous space [Mikolov et al., 2013], research works in NLP have consistently resorted to the cosine metric to cluster words based on semantics. Whether such embeddings are "non-contextualized" i.e., learnt through simpler neural networks [Pennington et al., 2014] or "contextualized" by bidirectional attention mechanisms [Devlin et al., 2018], works such as [vor der Brück and Pouly, 2019], [Ethayarajh, 2019b] have shown the fidelity of the cosine similarity metric to gauge textual similarities thereby allowing relationships between texts or words to be expressed geometrically.

3.3.1 The Cosine Function

The cosine similarity function for calculating the semantic similarity between vector embeddings is mathematically defined as follows:

$$Similarity(\mathbf{p}, \mathbf{q}) = \cos(\theta) = \frac{\mathbf{p} \cdot \mathbf{q}}{|\mathbf{p}| \cdot |\mathbf{q}|} = \frac{\sum_{i=1}^n \mathbf{p}_i \mathbf{q}_i}{\sqrt{\sum_{i=1}^n (\mathbf{p}_i)^2} \sqrt{\sum_{i=1}^n (\mathbf{q}_i)^2}}$$

¹⁷Note that that embeddings we generate from Transformer models, whether in their native state or affine mapped, are not *normalized*. Therefore, to maintain consistency as well as to focus on the directional aspects of the embedding vectors, we chose to not use Euclidean distance to measure semantic similarity.

Here, $\mathbf{p}$ and $\mathbf{q}$ represents our word and sentence embedding vectors and in the case of our second task of cross-document coreference resolution, even document vectors. For example, most language model embeddings for the task of cognate detection from models such as MBERT or Indic-BERT are 768 dimensional except for XLM-R which are 1280 dimensional. We have not resorted to any dimensionality reduction techniques like Singular Value Decomposition (SVD) or Principal Component Analysis (PCA) since our problem is one of supervised learning with sample sizes small enough to run classification models without needing excessive compute power on GPUs but large enough to avoid the problem of curse of dimensionality [Verleysen and François, 2005] in higher dimensional embeddings.¹⁸ Moreover, our intention was to study the semantic-transfer related properties of embeddings in their native state without injecting any perturbations arising from such dimensionality reduction methods.

3.3.2 Selection of Transformer-based BERT Variants

Even though cognates do not need to have similar meaning, many do share a semantic affinity [Kanojia et al., 2021b]. Work such as [Turton et al., 2021] suggest that contextual semantic information at the word level can be extracted from BERT and variants as embeddings. As such, we extract semantic information from both word-level and sentence-level embeddings from large multilingual Transformer-based models such as XLM-R [Conneau et al., 2020] and MBERT [Devlin et al., 2018], as well as from some smaller models, Indian language-focused models: IndicBERT [Kakwani et al., 2020] and Muril [Khanuja et al., 2021].

Below is a brief description of the five models we have selected. Our aim was to select a set of larger multilingual models like XLM and MBERT that supports the two target languages along with many other world languages as well as a set of smaller Indian-Language specific models. To support our hypothesis on semantic transfer from a mono-lingual model, we have also included a

¹⁸Our future experiments might involve such techniques to reduce dimensions when using pairwise scoring methods.

self-trained¹⁹ language model called AxomiyaBERTA. Their architectural details as well as their proven advantages as follows:

MBERT MBERT is the multilingual version of BERT that supports 104 languages²⁰ with a proven capacity for semantic knowledge transfer on downstream tasks [Pires et al., 2019]. While the general masked language modeling (MLM) and next-sentence prediction objective function in its training overlaps with classic BERT [Devlin et al., 2018] model, it has some differences: i) it includes parallel multilingual sentences in batch-wise training with MLM, and ii) its vocabulary is larger but shared among all languages and tokenized with WordPiece from its entire corpora. [Pires et al., 2019] show that MBERT’s ability to transfer is due to a multilingual representation, which enables it to manage transfer across different scripts. Its contextualized embeddings share a common subspace that contains linguistic information and is language-agnostic.

XLM-R XLM-R is a cross-lingual (100-languages) pretrained model which extends the previous MLM loss function with a Translation Language Modeling (TLM) objective and Causal Language Modeling objectives. It has demonstrated success in unsupervised machine translation tasks [Conneau and Lample, 2019]. XLM uses byte-pair encoding subword tokenization [Sennrich et al., 2016] which includes the most frequent symbol pairs when creating the token vocabulary. This makes it suitable for encoding tokens common in morphologically-rich and relatively agglutinating low-resourced languages (LRL) like Bengali and Assamese. This also alleviates bias towards high-resource languages, by reducing tokenization of LRL words at the character level. Moreover, this improves the alignment of embedding spaces of languages that share either the same alphabet or proper nouns [Smith et al., 2017], both of which can characterize the cognates in our datasets.

¹⁹This language model was trained at the Department of Computer Science, CSU and we intend to make it open-source in the future.

²⁰MBERT includes Bengali in its training data but not Assamese whereas XLM-R was trained with data from both languages but the Assamese training data size is significantly smaller with 5 million tokens. Bengali training data is over 100 times larger and the training data of a well-resourced language like English is 100 times larger still.

IndicBERT and MuRIL IndicBERT and MuRIL are focused on Indian languages and so have a larger relative training data size for languages like Assamese and Bengali. They also outperform XLM and MBERT against several semantic downstream NLP task benchmarks like IndicGLUE [Kakwani et al., 2020], cross-lingual XTREME [Hu et al., 2020], etc.

3.3.3 AxomiyaBERTA

For more enriched contextualized representations of Assamese semantics, and to investigate how much information a much smaller monolingual Transformer model might be able to transfer, we trained a "light" ALBERT (`albert-base-v2`) model for 305,700 steps with a vocabulary size of 32,000 on four publicly-available Assamese datasets: Assamese Wikidumps²¹, OSCAR [Suárez et al., 2019]²², PMIndia [Haddow and Kirefu, 2020]²³ and the Common Crawl (CC100²⁴) Assamese corpus [Conneau et al., 2020](in total, after preprocessing, around 14 million²⁵ Assamese tokens) with the BERT Masked Language Model [Devlin et al., 2018] loss function.

We have used the SentencePiece tokenizer [Kudo and Richardson, 2018] for sub-word tokenization since it is language-agnostic and is known to work well for low-resource languages. Table 3.4 gives configuration details of the monolingual Assamese Transformer model that we trained for this research. AxomiyaBERTA derives its name from "Axomiya" which is the native form of pronunciation for "Assamese" while "BERTA" derives from Assamese "barta" meaning "conversa-

²¹<https://archive.org/details/aswiki-20220120>

²²<https://oscar-corpus.com>

²³<https://paperswithcode.com/dataset/pmindex>

²⁴<https://paperswithcode.com/dataset/cc100>

²⁵In resource-scarce computational environments and especially for low-resource languages such as Assamese, it is challenging to find a huge monolingual corpus to train a Transformer model. For instance, the XLM-R which is a large language model was pretrained on about 2395 GB (164.0 billion tokens) out of which only 5M were Assamese tokens(iso code as). However, as suggested by [Ogueji et al., 2021], smaller datasets for low-resource languages can still work *better* than joint training with high-resource parallel corpora and we intend to explore if this is true for our AxomiyaBERTA model.

tion". Despite the name, AxomiyaBERTA is an ALBERT variant, not a RoBERTa variant, but also recalls the name Asom Barta²⁶, the official newsletter of Government of Assam.

Parameters	Config
architecture	AlbertForMaskedLM
attention_probs_dropout_prob	0.1
bos_token_id	2
classifier_dropout_prob	0.1
embedding_size	128
eos_token_id	3
hidden_act	gelu
hidden_dropout_prob	0.1
hidden_size	768
initializer_range	0.02
inner_group_num	1
intermediate_size	3072
layer_norm_eps	1e-05
max_position_embeddings	514
num_attention_heads	12
num_hidden_groups	1
num_hidden_layers	6
position_embedding_type	"absolute"
transformers_version	"4.18.0"
vocab_size	32001

Table 3.4: AxomiyaBERTA Model configuration trained on a monolingual Assamese corpus.

3.3.4 Word and Sentence Embeddings

Sentence-sensitive embeddings: For generating the sentence-sensitive embeddings²⁷ from cognates, we used the manually created sentences as described in Sec. 3.2.1 for further data processing. Our intention was to generate contextualized representations specific to the cognate word but within the context of the sentence in which it is present. We added two additional special tokens (<m> and </m>) to the models’ vocabularies. Before getting the sentence embeddings, the

²⁶<https://asombarta.com/>

²⁷Sentence-sensitive embeddings were generated only from MBERT and the AxomiyaBERTA, as the other models all have at least some support for Assamese already.

Language	Sentence	IPA
Bengali	এটি একটি টাং	eti ek̪i taŋ
Assamese	এইটো এটা ঠেং	eitu eta tʰɛŋ
English	This is a <u>foot</u> / <u>leg</u>	

Figure 3.1: Sample equivalent sentences with cognate words (and English translations) underlined.

cognate words (underlined in Fig. 3.1) were surrounded by these tokens to account for subword tokenization potentially breaking up the cognate words. This allows us to create binary vectors for the cognates using the indices of the special tokens in the sentence. Our model attends to these binary maps by an element-wise tensor multiplication in the forward function and outputs a contextualized representation of the word.

Word-level embeddings For each of the five models, we input a "sentence" formatted as [CLS]<word>[SEP] and use the [CLS] token’s `last_hidden_state` to get representations for each token in each sequence of the batch from the last layer of the model which often encodes more semantic information [Jawahar et al., 2019b]. The [CLS] token here serves the same purpose as the <m> tokens in the sentence-sensitive embeddings: to account for potential subword tokenization effects.

After extracting the embeddings from each model, we carry out the affine mapping (see below) of native embeddings in the form of a cartesian product combination between the various models.²⁸ Thereafter, we use the cosine similarity metric (defined in Sec. 3.3.1) to calculate semantic similarities between the word and sentence embeddings. These cosine similarities are input features into the final evaluation.

²⁸We create two labels i.e., `source` and `target` to map the native models. While word embeddings from a model are affine mapped to every other model, sentence embeddings are only mapped from MBERT to AxomiyaBERTA.

3.3.5 Affine Transformations Between Embedding Spaces

Only embeddings retrieved from the same model architecture are guaranteed to be directly comparable. In other words, a distinct word embedding preserves similarity relations across its dimensions only *within* that specific embedding space. Moreover, differences in training corpora, objective functions, optimization routines and model architecture mean that embeddings retrieved from different models are likely to be orthogonal in most dimensions.

However, recent work in the vision community [McNeely-White et al., 2020b, McNeely-White et al., 2022] has demonstrated that by fitting affine matrices $M_{S \rightarrow T}$ and $M_{T \rightarrow S}$ between paired features denoting equivalent samples extracted from a source model S and a target model T , features from one embedding space can be transformed to another embedding space with a high level of fidelity. This requires solving for a mapping function $f(x; W)$ where $W \in \mathbb{R}^{d_S} \times \mathbb{R}^{d_T}$, between equivalent information samples (i.e., paired embedding vectors) from two models, using a ridge regressor. The aforementioned work has been applied to CNN architectures, and here we use this task to explore the potential applicability of similar principles to Transformer architectures.

Affine mapping procedure Native model embeddings are often independently useful downstream NLP tasks, but their utility may be degraded when the language model does not robustly support the language in question. For instance, MBERT does not support Assamese and as such, many Assamese words may be subject to irregular sub-tokenization that do not capture the semantics of the original word. Therefore, we explore if and how linearly mapping one set of embeddings from its native space to a target model space can still transfer semantic information that is reflected in the classifier performance for the cognate detection task.

To construct the mapping, we take the word or sentence embeddings from one model as inputs, and equivalent word or sentence embeddings from another model as outputs, and fit them to each other using scikit-learn’s Ridge regressor. The resulting $d_S \times d_T$ transformation matrix²⁹ computed from a set of paired vectors serves as a *bridge* or transformation from one embedding

²⁹All embeddings used here are 768 dimensions, except embeddings from XLM-R, which are 1,280 dimensions.

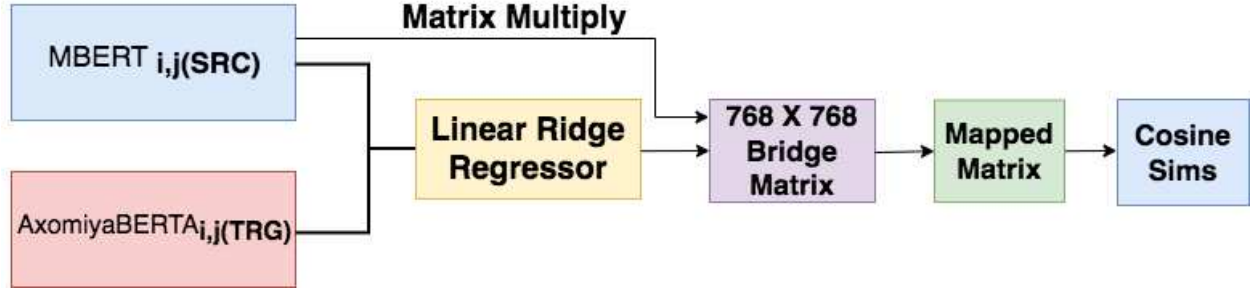


Figure 3.2: Cross-embedding space mapping pipeline resulting in directly comparable vector representations (MBERT→AxomiyaBERTA used as example).

space to another by minimizing the distance between paired points in $\mathbb{R}^{d_S} \times \mathbb{R}^{d_T}$ feature space that share equivalent semantics. Multiplying a source embedding by this precomputed bridge matrix should result in approximately the same semantics in the target embedding space, meaning that a transformed embedding and one native to the target (or source) embedding space are now directly comparable using metrics like cosine similarity. Fig. 3.2 shows this procedure.

3.3.6 Semantic Feature Exploration: A Sanity Check!

In order to understand whether and to what extent semantic information gets encoded in native model embeddings as well as affine-mapped embeddings, we now conduct a data exploration experiment with Kernel Density Estimation or KDE and Pair Plots to explore their relative contributions. KDE Plots are an amazing visualization tool that maps our cosine similarity-based features into a smooth probability density function. Since our data is continuous, this allows us to understand how various samples are encoded in either distinct or overlapping semantic spaces. On the other hand, Pair Plots help us visualize bi-variate pairwise analysis of the cosine similarities while their diagonals represents the univariate marginal distribution of all the models (native or mapped) involved.

Native Embedding Spaces: Encoding We get native word (not sentence) cosine similarity KDE Plots and Pair Plots for all the four models except AxomiyaBERTA since the latter does not support Bengali.

These plots give us an interesting insight into how semantic information is encoded for the four types of samples (cognates, synonyms, hard-negatives and random pairs). On average, all the native model embeddings capture or encode semantic information consistent with our assumptions of sample types. For instance, we assume cognates and synonyms to have high semantic similarity compared to hard-negatives and randoms. Most native models encode cognates and synonyms in a distinct semantic space more toward a value of cosine similarity of 1 as shown in Fig. 3.3 while the other samples probability distribution falls to the left of the plots. Interestingly, cognates are encoded in a still more distinct semantic space compared to synonyms even if both are assumed to be similar in meaning. This suggests that language models in their native state differentiate between apparently similar meaning word pairs perhaps because a contextualized attention mechanism sees more cognates paired with other cognates in its training samples. In other words, BERT-type encoders in their native state can more effectively "space out" cognates than synonyms while *still* encoding semantic information for both.

Another interesting trend is that larger multilingual models like XLM-R and MBERT are more effective in the former phenomenon i.e., at "dispersing" vectors in the space more than smaller models like Indic-BERT or MuRIL as shown in Fig. 3.3. This observation is similar to [Ethayarajh, 2019a] which suggests that larger multilingual models encode semantics in a relatively more isotropic "wider" space than the smaller models. On the other hand, the smaller models tend to capture semantics in a "tighter" high dimensional cone for the last hidden layer. The latter phenomenon for the smaller Indian-language specific models is unexplored and interesting because not only does this hint at differences in relative semantic information encoding capacity between models but also is a potential opportunity to try other loss functions for smaller models' training to widen this space to allow *cross-lingual* semantic information transfer. The Pair Plots (Fig. 3.4) with diagonal histograms as well as the bivariate pairing in the respective axes for the larger models substantiate the above observations. In the next paragraph, we will talk about how this affects semantic transfer between source and target models.

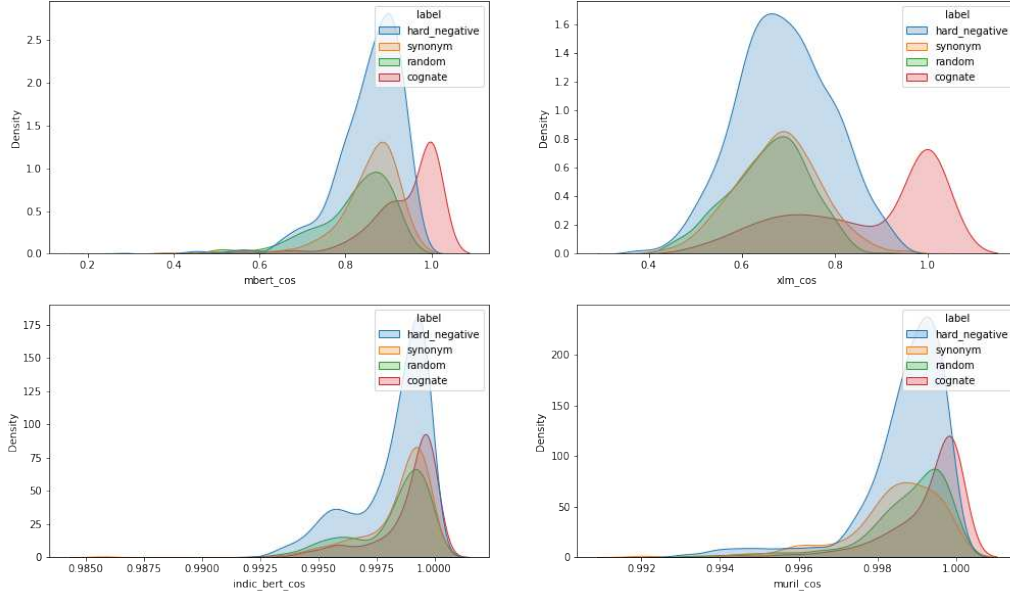


Figure 3.3: Kernel Density Estimation Plot for semantic similarity between word pairs generated as native embeddings from each of the four models, colored by pair type.

Affine Mapped Embeddings: Transfer Linear maps³⁰ from native models to target embedding spaces for words and sentences help us further distinguish the extent of semantic information transfer between the specific models in question. On average, as seen in Fig. 3.5, cognate probability distribution still lies more towards a higher cosine similarity indicating semantic transfer for cognates between certain specific models. The rest of the models clearly do not show such semantic transfer. We observe that affine maps between larger multilingual models like XLM-R and MBERT still encode cognates in a *distinct space* with higher cosine similarities. For instance, MBERT→XLM and vice versa as shown in the top-left of Fig. 3.5 helps us visualize this phenomenon. Interestingly, maps from large models (source) to smaller models (target) display this effect too but to a lesser degree except for when AxomiyaBERTA is the target space. In this case, the cognate cosine similarity falls in the middle of the kernel density distribution without a distinct semantic separation. However, the reverse case for AxomiyaBERTA (AxomiyaBERTA→XLM,

³⁰We use right arrow → to signify affine-maps from source to target embedding spaces.

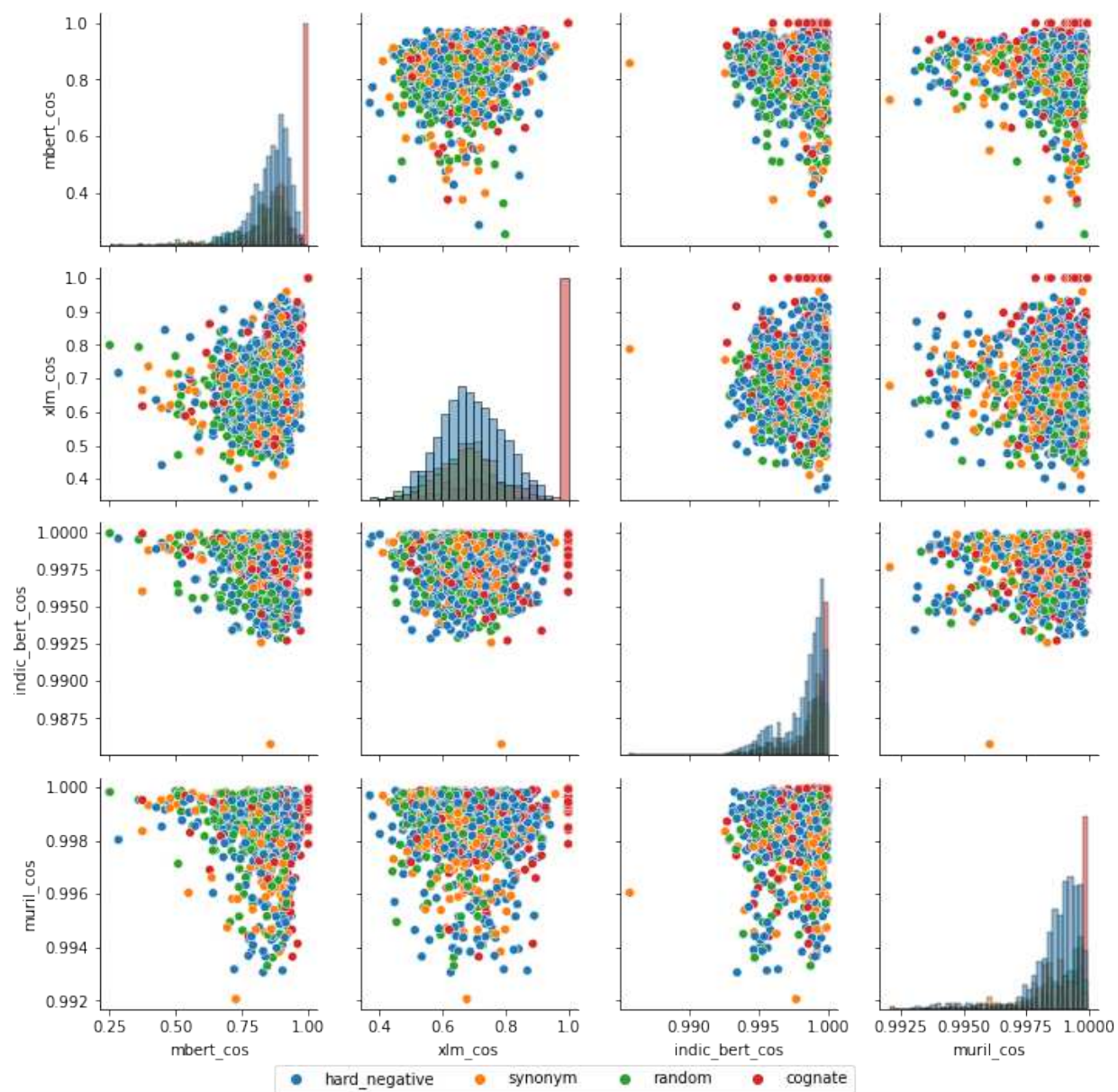


Figure 3.4: Pairwise Plots for semantic similarity between word pairs generated as native embeddings from each of the four models, colored by pair type.

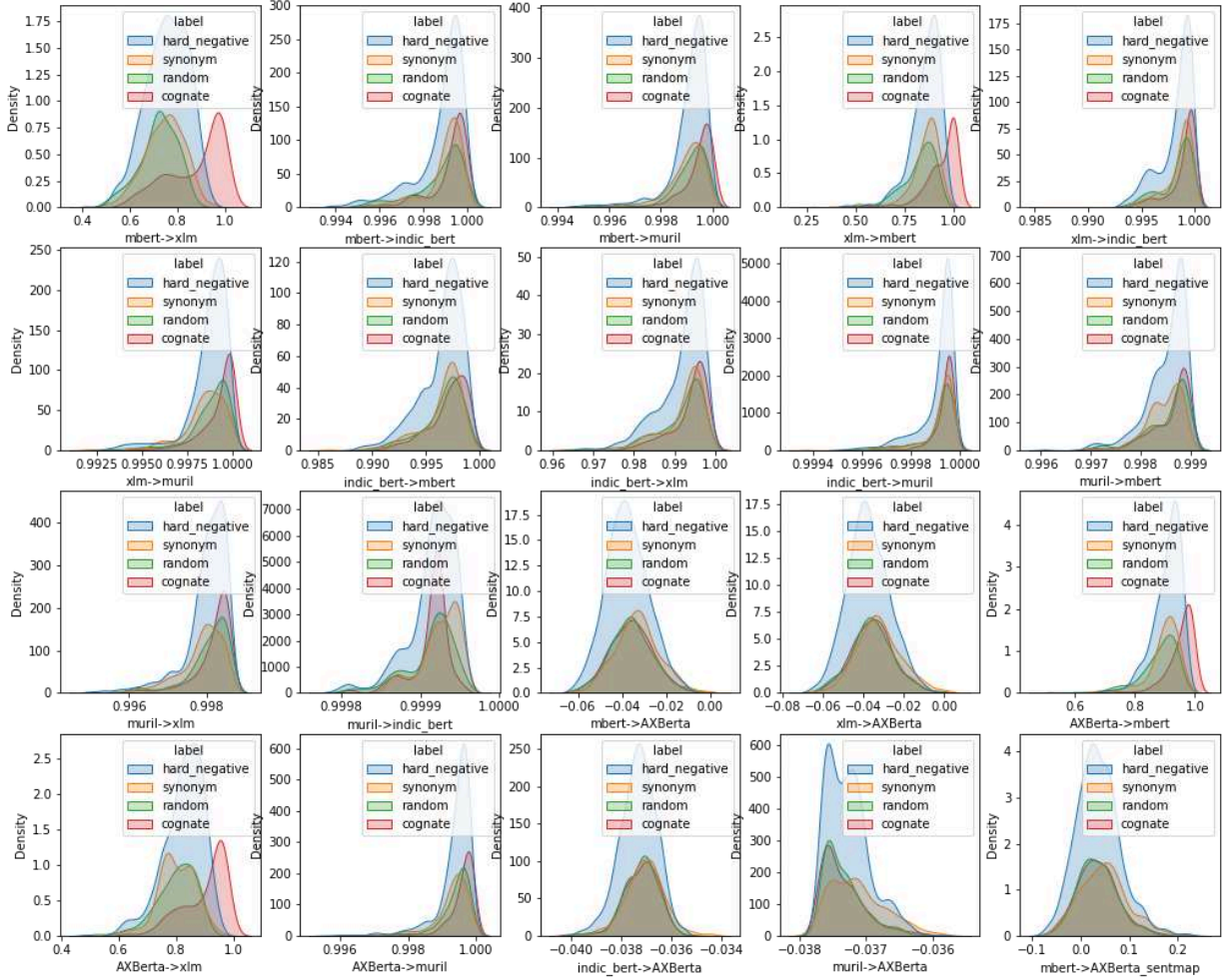


Figure 3.5: Kernel Density Estimation Plot for affine-mapped Word and Sentence Embeddings. Note: AxomiyaBERTA has been shown as AXBerta to fit the subplot legend.

where it acts as the source space) indicates a total volte-face. In this case, the extent of a distinct semantic space for cognates revealing a high cosine similarity is almost equal to native embeddings for the large models, slightly more so for XLM than MBERT. A little more digging reveals that since AxomiyaBERTA is monolingual, the native model encodes richer semantics at a word level without any cross-lingual perturbation while still transferring comparatively more semantic information to XLM since XLM already supports Assamese while MBERT does not. Similar trends can be observed from the corresponding Pair Plots (Fig. 3.6) for all mappings.

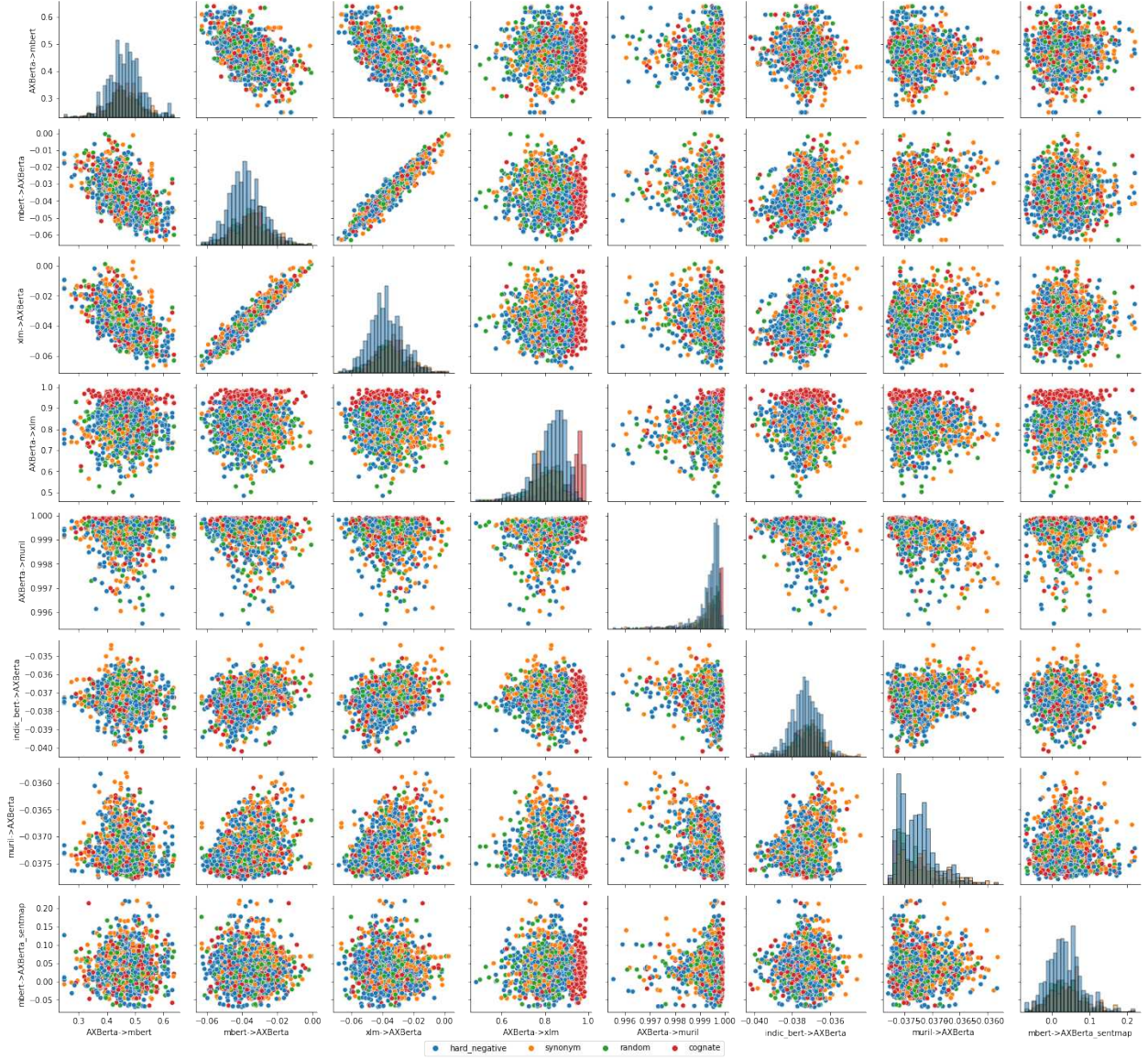


Figure 3.6: Pairwise Plot for affine-mapped Word and Sentence Embeddings. Note: AxomiyaBERTa has been shown as AXBerta to fit the subplot legend.

3.4 Non-semantic features

Although the primary focus of my thesis is the relative contribution of semantic features for this task, I also explore the contribution of non-semantic features (articulatory, orthographic and phonetic) to be able to gauge the relative importance of the former. In this section I briefly discuss the methodology of getting the non-semantic features as well as the resources and the neural network models used in the generating these features.

3.4.1 Articulartory features: Alignment Network

From a linguistic perspective, Assamese and Bengali have crucial differences in phonotactics, epenthesis, elision, and metathesis. Such differences naturally lead to differences in articulation between the two. Therefore, we build a model to align phonemes in the pair to capture articulatory characteristics native to a language but more importantly, one that attends to the *directional flow* of articulation between a pair. We hypothesize that this will provide a more informative measure than phonetic features like edit distances. [Mortensen et al., 2016] show that information-rich phonological representations do better than character-based models or one-hot encodings in NER tasks. PanPhon³¹ was used to convert the IPA transcriptions to 21 subsegmental articulatory features. These features include place and manner of articulation, voicing, etc., and the feature vectors were padded with zeroes to the maximum length of a vector in the cognate pair. The individual padded vectors were then concatenated for input to the alignment-scoring network.³² One point of note is that the alignment network is not to predict cognate status directly, since we do not include any semantic information at this step. However, the label acts as a rough indicator of how phonetically aligned the word pairs are. We only use the pre-softmax logits from this network for feature extraction. We hypothesize that there should be a positive correlation between the pre-softmax logit scores that we generate from this network at inference time to how strongly the sample word pairs are phonetically aligned.

3.4.2 Orthographic and Phonetic Edit Distances

Orthographic similarity is simply the Levenshtein edit distance [Levenshtein et al., 1966] between two strings. Since Assamese and Bengali use the same scripts with small modifications, we want to explore the importance of a simple string similarity metric as a feature in our classifi-

³¹PanPhon does not contain suprasegmental or tonal information but both Bengali and Assamese are non-tonal languages. So, we are safe here!

³²A two-layer deep feedforward neural network with 512 neurons in each layer, all with ReLU activation and followed by 10% dropout. We trained for 5,000 epochs on the aforementioned concatenated features of the All-languages dataset (see Sec. 3.2), using a 80:20 Train/Validation split. The network was trained against the cognate/non-cognate binary label.

cation task. This has been a oft-used feature in previous works in cognate detection and for our ablation studies, it also serves as a good *baseline feature* to compare the importance of semantic features. Notwithstanding the orthographic overlaps, for a balanced feature set, we need to extend our feature set to include some phonetic features too. As such, we calculate 6 different edit distances from PanPhon over the IPA transcriptions of the word pairs in our datasets. These edit distances are: Fast Levenshtein Distance, Dolgo Prime Distance, Feature Edit Distance, Hamming Feature Distance, Weighted Feature Distance, Partial Hamming Feature Distance, all normalized by the maximum length of the two words in the pair. We hypothesize that these distance metrics collectively capture some important information about phonetic similarity between Assamese and Bengali cognate pairs. The Fig. 3.7 shows the entire pipeline including all the feature types used for the task of cognate-detection.

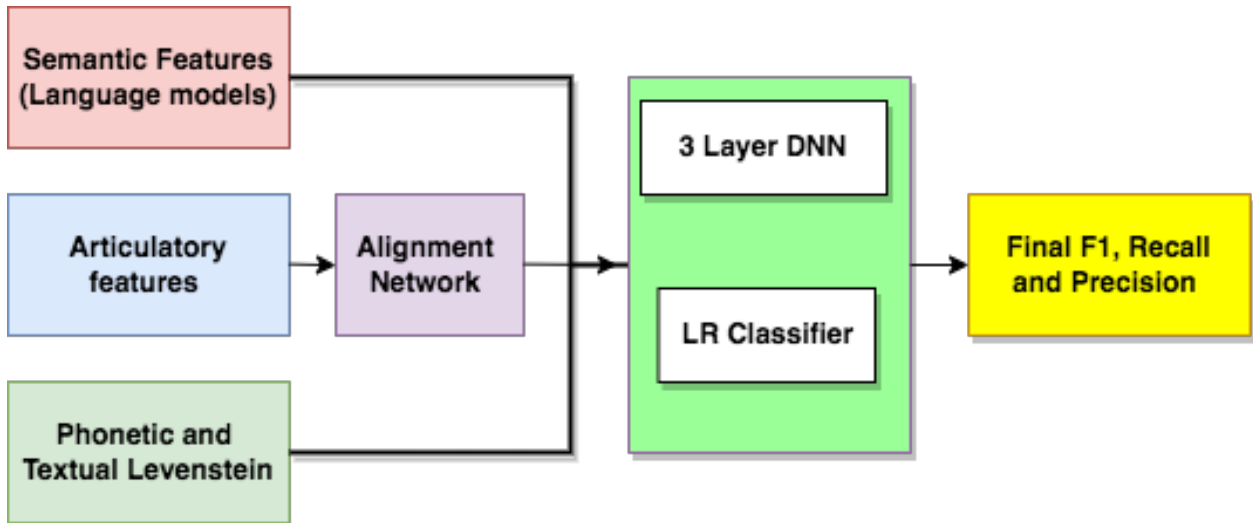


Figure 3.7: Entire Pipeline for Cognate Detection Task with Phonetic, Articulatory and Semantic Features. Note that the Alignment Network and the 3 Layer DNN are two separate neural networks, the former being used for getting alignment scores while the latter for classification.

3.5 Evaluation and Analysis

3.5.1 Classifier Models and Feature Abbreviations

With the variety of phonetic, semantic, and articulatory alignment metrics that we have extracted for all the paired words in our datasets, we now train classifier models on these features to discriminate cognates from non-cognates in the data, using these features. We train two types of classification models: a Logistic Regressor (**LR**)³³ and a Neural Network (**NN**). The NN consists of 3 layers of 512, 256, and 128 hidden units respectively, all with ReLU activation and followed by 10% dropout, and a final sigmoid activation, and is trained for 5,000 epochs with Adam optimization and BCE loss.

More specifically, we train three versions for each classifier: one trained on the `All-languages` dataset, and evaluated on the test splits of that dataset and of the `Assamese-Bengali` and `Bengali-Assamese` datasets; and one each trained and evaluated only on the `Assamese-Bengali/ Bengali-Assamese` datasets. We call the latter as pair-specific models which are hereafter denoted in tables and charts with an asterisk (*) or additional label `train_ev`). We trained all classifiers multiple times using different feature combinations to assess the contribution of different types of features. Table 3.5 shows the **feature abbreviations** we use in the following discussion for the different classes of features.

3.5.2 Influence of Features

I build the case for the importance of semantic features by first establishing non-semantic baselines. These baselines could be an orthographic, phonetic or even a subset of the semantic feature space so that we can compare this baseline with the semantic feature that is being ablated against. We have also included the overall best performance results and its feature set details to compare and contrast. Our method is one of error analysis i.e., we evaluate the results and then check for misclassified samples in our ablation studies to understand whether and to what extent

³³The LR is more interpretable since it allows us to easily analyze its regressor weights but the NN is better performing. Therefore, we mostly use the NN to compare and contrast various feature based results while the LR helps us analyze how much each feature is contributing to that result.

Abbr.	Features
ped	Phonetic Edit distances (PED)
dl	DNN logits (alignment score)
ed	PED with textual Levenstein dist.
b	All native MLMs (BERT variants)
m	All mappings w/o native MLMs
ab-am	All MLMs w/ word-level maps
ab-sm	All MLMs with sentence maps
sm	Sentence maps

Table 3.5: Abbreviations for feature combinations.

*sm - sentence maps from MBERT to AxomiyaBERTA space.

*b - includes native MLM embeddings but does not include any cross-embedding space mappings (word or sentence). Note that MLMs here mean multilingual language models and not the masked-language model loss previously mentioned with the same abbreviation in this thesis.

*ab-am - includes native MLM embeddings along with word embedding maps without sentence maps.

semantic encoding and transfer results from our affine mapping technique between embedding spaces. We also add a layer of interpretability to the results by cross-checking against the weights assigned to the different features by the LR classifier.

Overall Best Case

Table 3.6 shows positive Precision, Recall, and F1 for the neural network (NN) classifier using a combination of all the features (ed-dl-ab-am). We achieve high overall performance on cognate detection, with 94% F1, 93% recall, and 95% precision. We can also see that the classifier performs very slightly better using Bengali as the baseline language than using Assamese. Similar results hold for other feature subsets: using the "bidirectional" All-languages model, feature sets ed-dl-am, ped-dl-ab-am, and ed-dl-ab all show 94% F1(+) for Bengali-Assamese but 93% F1(+) for Assamese-Bengali.³⁴ We also see that the model trained on the bidirectional data outperforms in each direction models trained on that direction alone.

The NN classifier outperforms the LR by $\sim 4\%$ in all metrics. This suggests that for detecting bilingual cognates using multiple feature types, the non-linear decision boundary of a multi-layer perceptron system is better-suited to this task than the linear decision boundary of the LR.

³⁴One possible reason for this is that Bengali forms are on average somewhat more *conservative*, tending to preserve consonant clusters more than Assamese. In fact if we look at the false negatives for this result, we find many cases where one cognate has a consonant cluster and the other does not (see Table 3.7). Another possible reason may be the slightly higher number of Bengali baseline pairs in the dataset (see Sec. 3.2).

	<i>all</i>	<i>bn-as</i>	<i>as-bn</i>	<i>bn-as*</i>	<i>as-bn*</i>
P(+)	95	97	94	90	90
R(+)	93	94	92	88	87
F1(+)	94	95	93	89	88

Table 3.6: NN classifier results (as %) for the ed-dl-ab-am feature combination (full feature set).

Bengali	Assamese
জাঁঝ (/ʃädz ^{fi} /)	জন্দিয়া (/xond ^{fi} ija/)
শিক্ষা (/ʃikk ^{ha} /)	শিকোৱা (/xikɔwa/)
মিটি (/mifti/)	মিঠা (/mit ^{ha} /)

Table 3.7: Sample false negatives.

Alignment Features

The alignment score (dl) is the clear winner among all feature types since it increases performance by most, on average. (Table 3.8). Adding alignment scores to just edit distances (ed) causes performance to rise approximately 17%. The Logistic Regressor for the ed-dl feature set gives the alignment score feature a weight of ~ 3.2 , making it strongly correlated with cognate status. It also performs best using the bidirectional data; with addition of alignment score, the pair-specific models perform about 4-6% lower.

Feat.	<i>all</i>	<i>bn-as</i>	<i>as-bn</i>	<i>bn-as*</i>	<i>as-bn*</i>
ed	76	76	76	76	76
ed-dl	93	93	92	86	88
ped	43	43	43	42	51

Table 3.8: F1(+) as % with and without alignment score (dl) and Levenshtein distance features for NN model

Phonetic vs. Orthographic Features

When using only phonetic edit distances (ped), performance drops to 43% F1 in most evaluations (51% on the Assamese-Bengali pair-specific model). This is because many times Assamese-Bengali cognates are pronounced differently even if spelled similarly. Adding a textual Levenshtein distance metric (ed) can identify correspondence where phonetic edit distance struggles.

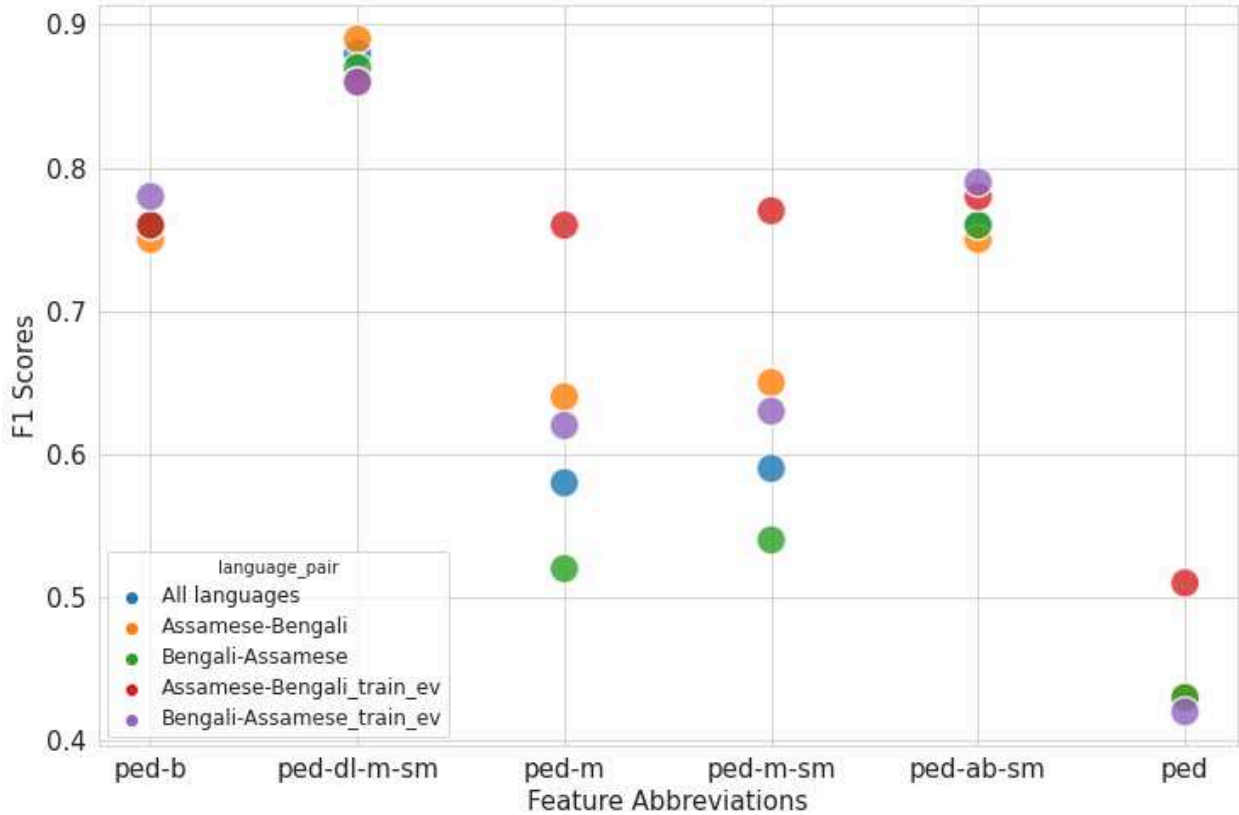


Figure 3.8: Influence of different semantic feature sets compared to phonetic edit distance baseline (*ped*).

The *ed* LR classifier gives textual Levenshtein distance a weight of ~ -2.7 , a strong inverse correlation.

Semantic Features

Our experiments show that while semantic features on average are not the top performing feature type, they do add much more nuance to the results. We find that semantic transfer between embedding spaces can be accomplished with affine maps and we substantiate it by studying individual word and sentence maps. To sum up, we find that adding semantic features by affine transfer helps classifiers like NN and LR to disambiguate false-positives and false-negatives thereby giving slight boosts to performance. For instance, addition of all the available semantic features to the *ed-dl* feature set results in a performance boost of only a few percentage points (cf. Tables 3.6 and 3.8). Nonetheless, by conducting further ablation tests, we can show where the semantic features actually provide important information.

Fig. 3.8 shows the effects of different subsets of semantic features—cosine similarities between native MLM embeddings, and between embeddings mapped from AxomiyaBERTA to each MLM embedding space at the word and sentence level—compared to the lowest performing feature set, phonetic edit distances.

Adding any semantic information to phonetic features alone substantially improves performance of the NN classifier on cognate detection. For instance, adding cosine similarities from the different pretrained MLMs (`ped-b`) brings performance back up to $\sim 76\%$, or on par with the inclusion of textual Levenshtein distance. For this feature set, XLM cosine similarity has the highest weight: ~ 1.0 , while MBERT cosine similarity is next: ~ 0.4 (MuRIL: ~ 0.3 ; IndicBERT: ~ 0.06).

Semantic Encoding in Native Models In terms of overall performance metrics, adding semantic similarly to phonetic edit distance is as good as adding textual edit distance, but the specific misclassified examples in each case are quite different. This is important because this answers the question of how do we really know that native model semantic information is crucial for the NN classifier. Table 3.9 shows the breakdown of false positives by type of negative example using these two different feature sets. We see that feature set `ed` has a much higher false positive rate, and also that in most cases when semantic information is used instead of textual edit distance, the proportion of false positives that are synonyms goes down, suggesting that including semantic information from MLMs improves cognate detection by mitigating misclassification of synonyms.³⁵

Semantic Transfer in Word-level Mappings Adding cosine similarities taken after affine mapping AxomiyaBERTA word-level embeddings into the embedding spaces of the MLMs (AxomiyaBERTA $\rightarrow$ MLMs, represented as `ped-m` feature set) also improves performance. However, the effect is more nuanced than when using native cosine similarities. For most data splits, the performance boost is not as pronounced (e.g., an appreciable but modest increase from 43%

³⁵The exception to this is in the `ped-b` feature set for the Assamese-Bengali pair-specific model (*as-bn**), where 60% of false positives are synonyms, pointing to the relative weakness of Assamese semantic representations in the MLMs.

	<i>all</i>		<i>bn-as*</i>		<i>as-bn*</i>	
	ed	ped-b	ed	ped-b	ed	ped-b
HN	18	12	12	11	6	4
Syn.	18	5	8	1	5	6
Rnd.	4	1	2	0	1	0

Table 3.9: Number of false positives using `ed` vs. `ped-b` feature sets broken down by negative example type (hard negative, synonym, random). Bidirectional and pair-specific models shown.

to 54% F1 on the bidirectional model evaluated against Bengali-Assamese data), but a dramatic increase in performance is seen on the Assamese-Bengali pair-specific model, where positive F1 rises to 76%, equaling the performance of the same model using the native MLM similarities.

When we examine the LR weights of these features for interpretability, we see that the weight assigned to cosine similarities between the mapped AxomiyaBERTA embeddings and Bengali XLM embeddings (AxomiyaBERTA→XLM) is ~ 1.0 while the equivalent weight for (AxomiyaBERTA→MBERT) is ~ 0.4 . These weights are nearly the same as those assigned to the native XLM and MBERT cosine similarities. When we carefully compare this observation with the semantic transfer through KDE Plot visualization (Fig. 3.5), we see that cognate-related semantics are indeed usefully transferred for the LR classifier to rely on. This and the similar NN performance indicate that these affine mappings are contributing the nearly the same level of information as native embeddings.

However, weights assigned to AxomiyaBERTA→IndicBERT and AxomiyaBERTA→MuRIL space are both close to 0. This may be due to the larger size of the MBERT and XLM training corpora. The resultant embedding vectors in MBERT/XLM space are more dispersed³⁶ or "spaced-out", and perhaps closer to isotropic [Ethayarajh, 2019a], whereas IndicBERT and MuRIL vectors appear to be clustered in a tight high-dimensional cone. This means there is more "space" in MBERT and XLM to transfer useful semantic information through techniques like affine mapping. This is particularly interesting in the case of MBERT, which did not train on Assamese data, yet

³⁶See Fig. 3.5 for a visual understanding through a cosine-similarity based KDE distribution.

the embedding space appears able to accommodate meaningful information from Assamese embeddings.

Semantic Transfer in Sentence-level Mappings In the case of using sentence-based affine maps, we see the implications of adding richer monolingual semantic information. One inspiring insight we get here is that smaller monolingual models like AxomiyaBERTA can transfer *language-specific semantics*. This helps in detecting cognates of that specific language in a pair which in this case is Assamese. How do we know that? Because MBERT→AxomiyaBERTA (ped-m-sm) shows a drastic decrease of misclassified hard-negatives (almost by one thirds) for Assamese-Bengali pair-specific model while the trend for Bengali-Assamese model is the reverse! Since hard negatives are semantically distant from their phonetic-neighbor cognates and are strong adversarial samples, the introduction of Assamese semantic information helps disambiguate the cognates from hard negatives based on language-specific semantics. One trend that sort of weakens our argument for language-specific semantic transfer is the slightly increased synonym false positives for Assamese-Bengali vs Bengali-Assamese, but overall, the NN classifier compensates this by eliminating random false positives, and further reducing false negatives. Table 3.10 and Table 3.11 below show the breakdown of false positives and false negatives by type of negative example for ped, ped-m and ped-m-sm.

From a performance perspective, there is also a slight boost in performance of the NN model when sentence-maps are included. The Assamese-Bengali pair-specific model reaches 77% F1. Adding sentence-level mappings alone to phonetic edit distances increases performance over ped by only ~6%; the combination of word and sentence-level mappings is what provides this final small boost to the Assamese-Bengali pair-specific models. When we add sentence mappings to ped-b (ped-ab-sm), we see that this time the two pair-specific models see an appreciable improvement from 76% to 78% (Assamese-Bengali_train_ev) and 79% (Bengali-Assamese_train_ev), suggesting that similarities computed after sentence-level mappings can help language-specific models more than language-agnostic or multilingual ones.

	<i>bn-as*</i>			<i>as-bn*</i>		
	ped	pm	psm	ped	pm	psm
HN	31	48	45	47	10	15
Syn.	0	4	4	6	8	6
Rnd.	0	7	2	0	2	0

Table 3.10: Number of false positives in pair-specific model outputs using *ped*, *ped-m* (*pm*), and *ped-m-sm* (*psm*) feature sets broken down by negative example type (hard negative, synonym, random).

	<i>bn-as*</i>			<i>as-bn*</i>		
	ped	pm	psm	ped	pm	psm
FN	212	140	138	182	106	100

Table 3.11: Number of false negatives (undetected cognates) in pair-specific model outputs using *ped*, *ped-m* (*pm*) and *ped-m-sm* (*psm*) feature sets.

To sum up, the trends in Tables 3.9–3.11 show that using native cosine similarities from models with relatively strong support for Bengali helps Bengali-Assamese performance, while adding mapped word and sentence-level embedding similarities help Assamese-Bengali performance by bringing in more Assamese-specific information through affine transformation. In other words, semantic encoding in native model embeddings and *useful* semantic transfer through affine mapping: both occur.

It is to be noted that our approach for detecting cognates in the two similar languages is surely not without limitations. Since a major part of feature extraction criteria/process (phonetic, articulatory and semantic) was targeted to suit our task for two *similar* languages, we might get very different results for two relatively different languages. For instance, we expect that the interchangeability of feature types could work less well for two languages that are quite distantly-related like, for instance, English and Assamese. While cognates in such distantly-related languages might still exhibit semantic affinity, they might not display immediate phonetic affinity or articulatory alignment for our current method to work as it is. For instance, the Assamese word (IPA notation) for wheel is /sɔka/ which is cognate with the English word "wheel". They are semantically similar but phonetically quite different. In such cases, our method needs to be enhanced by either weighing the various feature types according to their importance for that language-pair (such weighing parame-

ters can be learned) or we might have to experiment with individual feature sets first before trying a combination of all the three feature types. Such experiments with distantly-related languages can surely be a way to test the robustness of our method for detecting cognates.

Chapter 4

CROSS DOCUMENT COREFERENCE

RESOLUTION (CDCR)

4.1 Background

Look at these two compound sentences below:

*[Victoria Chen]₁, CFO of [Megabucks Banking]₂, saw [her]₁[pay]₃ jump to dollar 2.3 million, as the [38-year-old]₁ became the [company]₂'s president. It is widely known that [she]₁ came to [Megabucks]₂ from rival [Lotsabucks]₄*³⁷.

The bold-faced words or phrases refer to some *real-world* entity, organization, person or an event. In the discourse of coreference [Jurafsky, 2000], each of the colored words is called a "mention". In NLP, coreference resolution is the task of determining whether two mentions "corefer" i.e., they refer to the same entity in the discourse model or a text. For instance, in the sentence shown above, words or phrases with the same numbered subscripts are coreferring. The set of coreferring expressions is often called a "coreference chain" or a "cluster". Therefore, the mentions³⁸ in each *same* subscripted set represents those coreferring mentions that need to be "linked" in a cluster.

Coreference Resolution (CR) Problem: Given a text T or a set of Documents D ³⁹, a CR system or algorithm has to find all the entities and the coreference links between them. Task performance is evaluated by comparing the *links* predicted in the system with those in human-created *gold coreference* annotations on T or D . In popular coreference evaluation systems, the input to the system is the raw text of articles, and systems must detect mentions and then link them into clusters, a technique we apply in our experiments too. However, in this thesis, I apply a cosine similarity

³⁷<https://web.stanford.edu/~jurafsky/slp3/21.pdf>

³⁸The word Lotsabucks colored in violet is a singleton mention since it only occurs once.

³⁹Coreference Resolution for a set of documents is usually known as the Cross-Document Coreference Resolution (CDCR) problem. Our experiments are on this task.

metric to score the set of documents in a pairwise fashion and then cluster the scores using an affinity matrix.

4.2 LDC Phase 1 Dataset and Knowledge Graphs

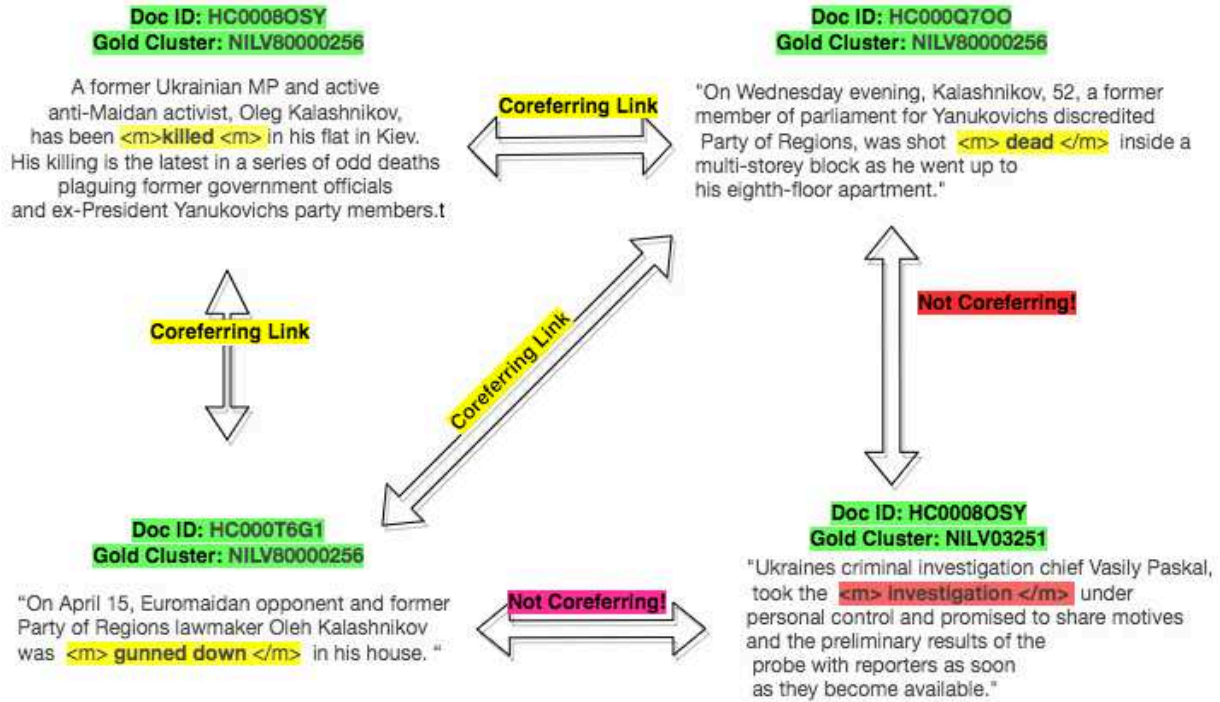


Figure 4.1: Example showing coreferring and non-coreferring links for events in LDC Phase 1 Dataset.

For our experiments, we use the LDC Phase 1 Dataset. The Linguistic Data Consortium (LDC) Phase 1 Dataset [Cieri et al., 2020] for events and entity and their common relations was created and released as a part of the DARPA AIDA (Active Interpretation of Disparate Alternatives) program. This project funds NLP and artificial intelligence-based research for developing semantic engines to distinguish between alternative interpretations of situations, events, and patterns while focusing more on unstructured data sources to gain insights into such data in noisy or deceptive conditions. For the purpose of this thesis, we evaluated our affine (linear) mapping-based CDCR (Cross Document Coreference Resolution) system using solely text-based data (including English, Russian, and Ukrainian languages) describing a specific set of annotated events and entities around

the 2014 Ukrainian war discourse. Most of the initial annotated data, as well as source data, were highly structured data. They were relatively uniform XML files with token-level information about the events and entities, along with meta-data about sentence separations, token types, etc., such that its structure can be mapped to its original text on the websites from which the event and entity data were extracted.

The gold annotations that contains the actual coreferring links were manually annotated as a part of the DARPA AIDA project. They are available as source files that can be used to map each event or entity to its respective gold cluster identification number. As such, if two mentions have the same gold cluster identification number (ID), they are coreferring. Some feature-based details and distribution events and entity links of the LDC dataset are shown in the Table 4.1 below. Note that the total number of entities are more than four times as that of events.

Features	Events	Entities
Total docs	69	70
Total mentions	847	3796
Unique gold clusters	271	1216
Max. cluster mentions	76	188
Unique lexical	405	1590
Singletons	198	685

Table 4.1: Table showing distribution of various features of the LDC dataset for events and entities. Note that the label "Unique lexical" represents lexically unique mentions. For instance, if "murdered" as an event mention occurs twice in the entire dataset and they do not corefer, then their respective gold cluster IDs will be different. Also, in that case, "murdered" will only be counted once in the "Unique lexicals" feature. Similarly, if a gold cluster occurs twice, it is only counted once in the Unique gold clusters label.

In order to get a better idea of how these features are represented, Figure 4.1 shows a set of four events from the LDC Phase 1 dataset, with their respective gold cluster IDs, Document IDs, the sentence text that contains the coreferring or non-coreferring mention text as well as the special tokens⁴⁰ (<m> and </m>) that surround the event mention text. We can see from our understanding of language that the three event mentions (in yellow surrounded by special tokens)

⁴⁰More details of our data-preprocessing steps and the role of the special token in the section on Transformer-based Bi-encoders.

within the sentences actually refer to the same real world event i.e., the death or the killing of the Ukrainian political-activist named "Kalashnikov". Things that help us identify or link this include political phrases like "Ukrainian MP", "opponent", "government officials" that are part of the three texts pieces involved. Proper nouns such as "Kalashnikov" further narrows our cognitive search space for identification while phrases like "flat", "apartment" or "house" help us reasonable identify the place or location of the event and thereby conclude that those three events to be coreferring. The same cannot be said about the fourth event on "investigation" which per se does not signify that specific death of the political activist.

Intuitively, for a human to link these three coreferring mentions (or unlink the fourth) requires a contextual understanding of the text and discourse-based characteristics. In this case, they are political phrases, proper nouns, places of residence, etc., that help us restrict out search space. However, such auxiliary context could reside beyond the sentence that spans the mention in a document. Therefore, the task of identifying coreferring event mentions (or entities) using artificial intelligence becomes all the more challenging due to the mentions not only being located at arbitrary positions in the document, sometimes even a few sentences away, but also being located across documents like in the case of CDCR. Another challenge is that simple *lexical overlaps* that might help identify coreferring links within a document cannot be trivially applied across documents since the lexically identical events or entities might not be coreferring since documents are authored independently [Cattan et al., 2021b]. Our idea, as such, is to use the context of the textual information available for each event or entity mention to decipher which ones need linking or are coreferring. We use Transformer-based language models for encoding such contextual and semantic information.

Knowledge Graphs Our first task was to extract the tokenized information from the XML files using XML file parsers like BeautifulSoup⁴¹ to create the knowledge graphs. Knowledge Graphs⁴²

⁴¹<https://pypi.org/project/beautifulsoup4/>

⁴²<https://www.turing.ac.uk/research/interest-groups/knowledge-graphs>

(KGs) organize data from multiple sources, capture information about entities of interest in a given domain or task (like people, places, or events), and establish connections between them. In data science and AI, knowledge graphs are commonly used to: map features in machine learning models, find links including the depth of various entities or features, and overall, as bridges between structured data and human understanding such as generating human-decipherable explanations.

In our task, we create two knowledge graphs, the *Document-Sentence Map* and the *Mention Map*, the former a single knowledge graph for both mention types and the latter, one for each individual mention type. Both of these are ordered nested dictionaries created in Python. The Document-Sentence Map allows us to correctly map each document with their document identification number (`DOC_ID`) along with the tokenized sentences that are a part of the document. This map also contains extra meta-data like sentence identification numbers, token-lengths of each token or word, etc. On the other hand, the Mention Map links event and entity specific information for each mention in the entire set of documents with their respective documents, `Mention_IDs`, start and end character of the mentions, gold cluster information, their `topic_IDs` etc. More importantly, for the purpose of our pairwise cosine similarity clustering that uses document level information for generating embeddings, the Mention Map helps us track down the sentence level as well as document level context of each mention, for both events and entities. For instance, as shown in Table 4.1, the 847 event mentions and 3,796 entity mentions can be easily accessed on a sentence or a document level from the event Mention Map to generate their respective document (or sentence) embeddings from a language model. More specifically, as shown in Fig 4.2, the "bert_doc" dictionary key in the Mention Map contains the document-level textual information correctly linked to that specific mention ("gunshot") in our set of documents. Similarly, "bert_sentence" key gives us the sentence-level textual information for that event mention. In our experiments, we use document-level textual information instead of sentence-level to get richer contextualized vector representations of mentions from our selected Transformer-based language models. The section below provides details of model selection and architecture.

```
[{'m_id': 462,
  'mention_id': 'VMIC0015MVH.000003',
  'doc_id': 'HC00080SY',
  'start_char': 489,
  'end_char': 495,
  'gold_cluster': 'NILV80000256',
  'topic': 'E101',
  'lang_id': 'eng',
  'mention_text': 'gunshot',
  'sentence_start_char': 477,
  'sentence': 'caused by a gunshot,',
  'bert_sentence': 'caused by a <m> gunshot </m> ',
  'bert_doc': 'Series of 'bizarre suicides' & murders: Former Ukrainian MP shot dead in Kiev\nOleg Kalashnikov (image by http://glavpost.com)\n\u200bA former Ukrainian MP and active anti-Maidan activist, Oleg Kalashnikov, has been killed in his flat in Kiev.\nHis killing is the latest in a series of odd deaths plaguing former government officials and ex-President Yanukovichs party members.\nThe 52-year-old was found dead at his residence in Kiev on Wednesday evening.\nHis death was\ncaused by a <m> gunshot </m> ,\nthe Interior Ministry said in a statement announcing a police inquiry.\nUkraines criminal investigation chief Vasily Paskal, took the investigation under personal control an
```

Figure 4.2: Example showing a Mention Map (events) knowledge graph in LDC Phase 1 Dataset

4.3 CDLM Model and other variant selection

For our experiments, we selected four Transformer-based large language models: CDLM [Caciularu et al., 2021], BERT_BASE_UNCASED⁴³ [Devlin et al., 2018], BERT_LARGE_UNCASED [Devlin et al., 2019] and Coref_BERT [Ye et al., 2020]. These large language models more or less use similar Transformer-based self-attention mechanisms to form a contextualized bidirectional representation of the languages they were pretrained on using the masked language modelling approach. In recent years, such models have shown very promising results in several downstream NLP tasks such as syntactic tasks like span boundary prediction or next-sentence prediction or semantic tasks like question-answering, Named-Entity Recognition, etc. Even though the four models were trained in varied settings with different hyperparameters and the concomitant hundreds of millions of trainable parameters, recent research [Cattan et al., 2021a], [Caciularu et al., 2021] have used similar models in their ablation studies to explore how such encoder-only models fare in tasks like coreference resolution.

In our case, except for the BERT_LARGE model which encodes texts in 1,024 dimensions, the other three are 768 dimensional in their embeddings. This allows us to compare how an increase in dimensions of embeddings may or may not capture the semantic information in texts and to

⁴³Hereafter, both the uncased BERT variants (base and large) will be shortened to BERT_BASE and BERT_LARGE respectively. We have restricted our model sample space to only *uncased* versions of these models since we want to maintain consistency of tokenizers among all the models selected. Also, unlike entity recognition tasks, CDCR does not explicitly require such case-related constraints.

what extent, especially post affine-mapping. Moreover, our ablation space was narrowed down by considering two coreference-specific language models (Coref_BERT and CDLM) while also including general language models that were not trained specifically to link mentions in texts. While most of these models are computationally intensive to train since they use multiple GPUs or TPUs, they are readily available as pretrained or finetuned (to a specific task) on the Huggingface Library.⁴⁴ In the section below, we provide a brief outline of the four models chosen for our task, their specialities and why they were selected. We also provide our definition of "contextuality" for such language models and rank them according to how contextualized they are.

CDLM CDLM [Caciularu et al., 2021] is our default model and is the most contextualized. It improves upon the popular masked language modeling objectives of different BERT models by simulating its pretraining within a cross-document environment, i.e., it is pretrained to learn cross-document relationships from a large set of related documents. It does it by cleverly utilizing the local and global attention mechanism from the Longformer Transformer model [Beltagy et al., 2020]. More specifically, the Longformer model facilitates the encoding of much larger documents by using *global* attention to the event and entity mentions as well as the [CLS] token but applying local attention to the rest of the tokens of the sentence or document. This not only enhances the semantic localism of contextual embeddings generated for each mention but also reduces the computational cost by reducing the quadratic time complexity of off-the-shelf Transformer models to linear in the sequence length.

More specifically, Fig. 4.3 provides an intuitive understanding of the global and local attention techniques in the Longformer model. Fig. 4.3 (a) shows a typical self-attention mechanism that usual off-the-shelf transformer blocks use. Its time-complexity is $O(n^2)$ if n is the sequence length of a training sample. Fig. 4.3 (b) and Fig. 4.3 (c) show the Longformer-based sparse-attention technique that uses a sliding or a dilated sliding window (respectively) approach to self-attention. In these cases, the time complexity reduces to $O(n \times w)$ or $O(n \times l \times d)$ respectively where

⁴⁴<https://huggingface.co/>

w is the length of the sliding window, d is the length of dilation of that window and l is the number of encoder layers or the receptive field. All three of these are constant, but their combined multiplication can be a large value. Therefore, the dilated sliding window can still be very compute intensive and close to quadratic if the number layers are more. However, as shown in Fig. 4.3 (d), the Longformer uses the dilated sliding window technique for most of the tokens while using a full n-squared attention for a few selected tokens like the [CLS] or task-specific tokens. In other words, it applies global attention to a few task-specific tokens while applying local attention to the rest. In this way, it reduces its time complexity to almost $O(n)$ since the selected tokens for global attention are few and independent of the sequence length.

BERT_BASE This is the classic BERT language model [Devlin et al., 2018] that learns bidirectional contextualized representations of the English language by predicting masked words and next sentences in its training. Unlike CDLM or CorefBERT, this has not been pretrained within coreference-specific setting which is crucial because it lets us compare embeddings from this model to our default CDLM model and understand the extent and the nature of semantic transfer. In other words, this is a less-contextualized model and since it only encodes information in 12 layers, it is also relatively smaller than other similar models like BERT_LARGE. As such, this is our baseline model for comparison.

BERT_LARGE All things remaining equal, this is the big brother of BERT_BASE. In other words, the training corpora, the training loss functions as well as the hyperparameters are same as BERT_BASE but it encodes information in 24 encoder layers instead of 12 and with 1024 as the embedding hidden dimension. Apart from the greater compute power needed to generate embeddings from this model, it also allows us to explore the extent of semantic transfer with increased-dimensionality related perturbations upon affine mapping. While this model is more contextualized than BERT_BASE, it is still relatively less contextualized than CDLM or CorefBERT with respect to tasks involving coreference resolution.

Coref_BERT Long distance mentions cannot be effectively learned in off-the-shelf language models like BERT_BASE where the focus is mostly on the masked word in the sentence (randomly selected and not focused on event or entity mentions) or the next sentence prediction. Thus, their attention is mostly semantically or syntactically local. In CorefBERT, long distance connections between mentions that exist beyond the span boundary of a sentence are learned which helps in coreference resolution. Keep in mind, that this long distance is around 512 tokens which is still eight times less than that of CDLM.

In order to understand high-fidelity semantic transfer between relatively more or less contextualized models, we rank the above four models based on their *contextualized* nature. As such, we define the extent of contextuality of a model in this decreasing order of importance: 1) whether the model was pretrained in a cross-document setting using the special tokens, $\langle m \rangle$ and $\langle /m \rangle$, 2) whether the model was simply pretrained to learn to link coreferring mentions, 3) whether additional finetuning for coreference tasks was carried out and 4) the depth (number of encoder layers) of the model if other things are consistent. Therefore, the order is as follows: CDLM (default model) $>$ CorefBERT $>$ BERT_LARGE $>$ BERT_BASE, where ' $>$ ' means more or greater than. Our analysis of the extent of semantic transfer after affine-mapping is based on the above ranking of the relative contextuality of the models.

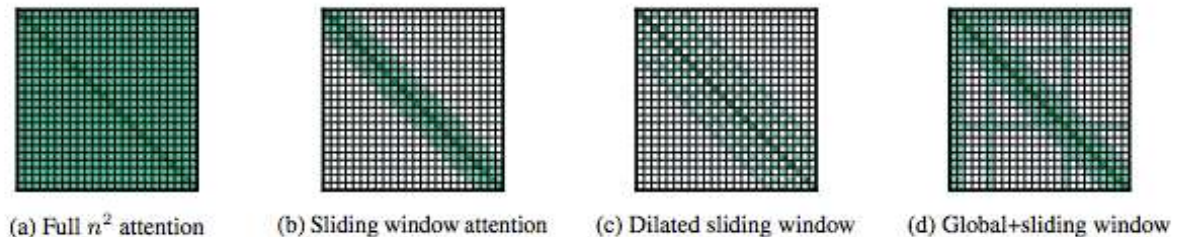


Figure 4.3: Global and Local attention that the CDLM model uses. [Beltagy et al., 2020]

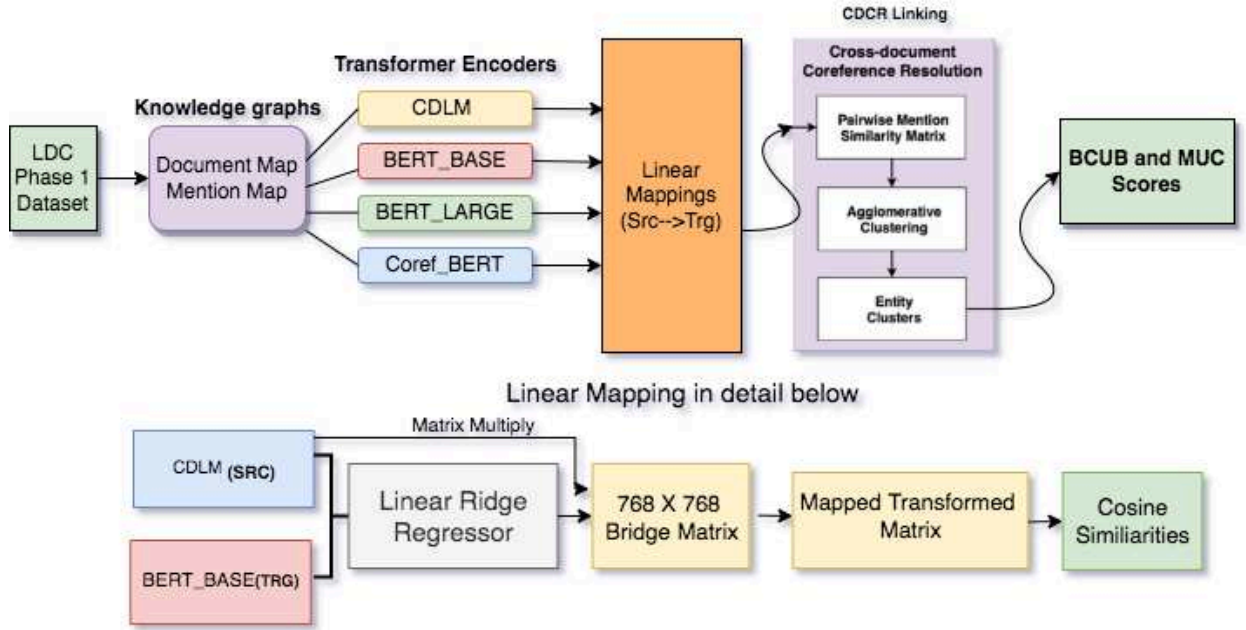


Figure 4.4: Entire CDCR Pipeline for Entities with details of the linear mapping technique in the segment below. Same pipeline applies for event mentions.

4.4 Bi-Encoder for Document-level embeddings

In order to create the contextualized source document embeddings for each mention, we surround the specific mentions in the documents with the additional special tokens that the Hugging-face library for these models allow us to use. This helps the model-specific tokenizers locate the position IDs of the mentions after the tokenization process. These position IDs (each of $\langle m \rangle$ and $\langle /m \rangle$ has a constant value) are made a part of the embedding space⁴⁵ as well as the vocabulary for these models. This is for the purpose of applying attention to the specific mentions during the forward function while getting the model output.

One of the challenges of encoding event or entity mentions with special tokens is that we have to ensure that the mentions do not occur beyond the maximum token-length of the specific Transformer model being used after the tokenization step. To solve for this edge case, we had to find the overflowing indices of the special tokens surrounding the mentions in the tokenized tensors

⁴⁵CDLM has been pretrained using these special tokens as a part of its embedding space while the other three are not; this means these special token embeddings are already learned in the case of the former while the rest were randomly initialized.

for each batch of documents if and when they lie beyond 512 tokens. Thereafter, for each of those overflowing indices, we cropped a 512-token span of the tensor⁴⁶ while ensuring that the special token indices are within that span for all the edge cases in a batch.

Since Transformer-based language models require inputs to be of a constant length in the last dimension for a batch of samples, we thereafter padded the overflowing tokenized indices to the max-length of the batch to avoid any batch-wise processing errors in token length. Our special attention mechanism then lets us create binary maps for each sample where the mention-specific indices are encoded as "1" (rest being "0"). We then apply an element-wise multiplication between the model embedding outputs of each sample with its respective binary map to get an overall semantic representation of the mentions within that sample document. This allows us to generate a contextualized embedding from the model as output. We call these embeddings as "native" since they are generated straight from the language models without any linear transformations applied on them. For more details about the edge cases and the algorithm being used for the CDCR bi-encoder we well as clustering, check out Fig.4.5.

Upon generating the contextualized embeddings from the chosen Transformer-based language models, we also carried out a Principal Component Analysis (PCA) projection of the CDLM embeddings to visualize the encoding of semantic information in our default model. This allows us to project the 768-dimensional embeddings into a 3-dimensional or 2-dimensional space while retaining 99 percent of the variation in the samples in the first dimension. This was done to explore and visualize whether contextualized semantic information relevant to our coreference resolution problem was *natively* encoded in our default model that can be amenable to high-fidelity semantic transfer.

⁴⁶In the case of CDLM we used its full capacity which is 4096 tokens thus adding more contextuality.

Algorithm 1 Cross-document Event/Entity Coreference Algorithm

Require: D :an ordered list of documents as dictionary
 M^e, M^v :gold entity and event mentions
 T : Cosine Similarity Threshold $\leftarrow 0.6, 0.75, 0.95$
 $CDLM, BERT_BASE, BERT_LARGE, Coref_BERT$: Pretrained Transformer Models
 $Vector_Maps_s, Linear_Maps_{s \rightarrow t} \leftarrow []$ (Native and Affine-mapped Embeddings for source (s) and target (t))
 $Cos^e, Cos^v \leftarrow []$ (Calculated cosine similarities)
 $m1, m2 \leftarrow []$ (Special token IDS)
for $i \in 1, \dots, |M^v|$ **do**
 if $i \in D$ **then**
 if $CDLM$ **then**
 $m1, m2 \leftarrow CDLMTokenizer(D, M^v)$
 if $m1$ or $m2 > 4096$ (max_token_length for CDLM Longformer) **then**
 $overflow_idx \leftarrow$ get overflowing indices
 $concat_tensor \leftarrow$ get concatenated tensor with span 4096 after padding
 $new_tensor \leftarrow$ final tensor with replaced tokens at $overflow_idx$
 $binary_maps \leftarrow$ generate binary map to special attention to mentions
 $Vector_Maps_s \leftarrow$ final model outputs after element wise multiplication with binary maps
 end if
 else if $BERT_BASE$ or $BERT_LARGE$ or $Coref_BERT$ **then**
 $m1, m2 \leftarrow BERTTokenizer(D, M^v)$
 if $m1$ or $m2 > 512$ (max_token_length for BERT Variants) **then**
 $overflow_idx \leftarrow$ get overflowing indices
 $concat_tensor \leftarrow$ get concatenated tensor with span 512 after padding
 $new_tensor \leftarrow$ final tensor with replaced tokens at $overflow_idx$
 $binary_maps \leftarrow$ generate binary map to special attention to mentions
 $Vector_Maps_s \leftarrow$ final model outputs after element wise multiplication with binary maps
 end if
 end if
 end if
 for $V_s^i \in |Vector_Maps_s|$ **do**
 for $V_t^j \in |Vector_Maps_t|$ **do**
 $Coeff_{s \rightarrow t} \leftarrow \text{Minimize}((V_t^j - V_s^i \beta)^T (V_t^j - V_s^i \beta) + \lambda \beta^T \beta), \lambda = 1 (\text{Assumption})$
 $Linear_maps_{s \rightarrow t} \leftarrow V_s^i \cdot Coef_{s \rightarrow t}$
 end for
 end for
 for $i \in 1, \dots, |M^v|$ **do**
 for j in $(i+1)$ **do**
 $Cos^v \leftarrow \text{Cosine Sims}(Linear_maps_{s,i}, Linear_maps_{s \rightarrow t,j})$
 $Similarity_Matrix \leftarrow \text{Csr_matrix}(Cos^v, T)$
 $Clusters, Labels \leftarrow \text{ConnectedComponents}(Similarity_Matrix)$
 $Scores \leftarrow \text{ConllScorer}(Labels, M^v)$
 end for
 end for
return $Scores$

Figure 4.5: Cross-document Event/Entity Coref Algorithm.

Fig. 4.6 (3-D PCA projection figure on the left) shows that event mentions like "protests", "military operations", "clashes" and "uprisings" are clustered in a relatively distinct space than events like "murder", "suicide" and "assassination". While the right figure, which is a 2-D PCA projection, clearly shows that death-like events like "died", "bludgeoned", "gunned-down" and "killed" are forming a distinct cluster.

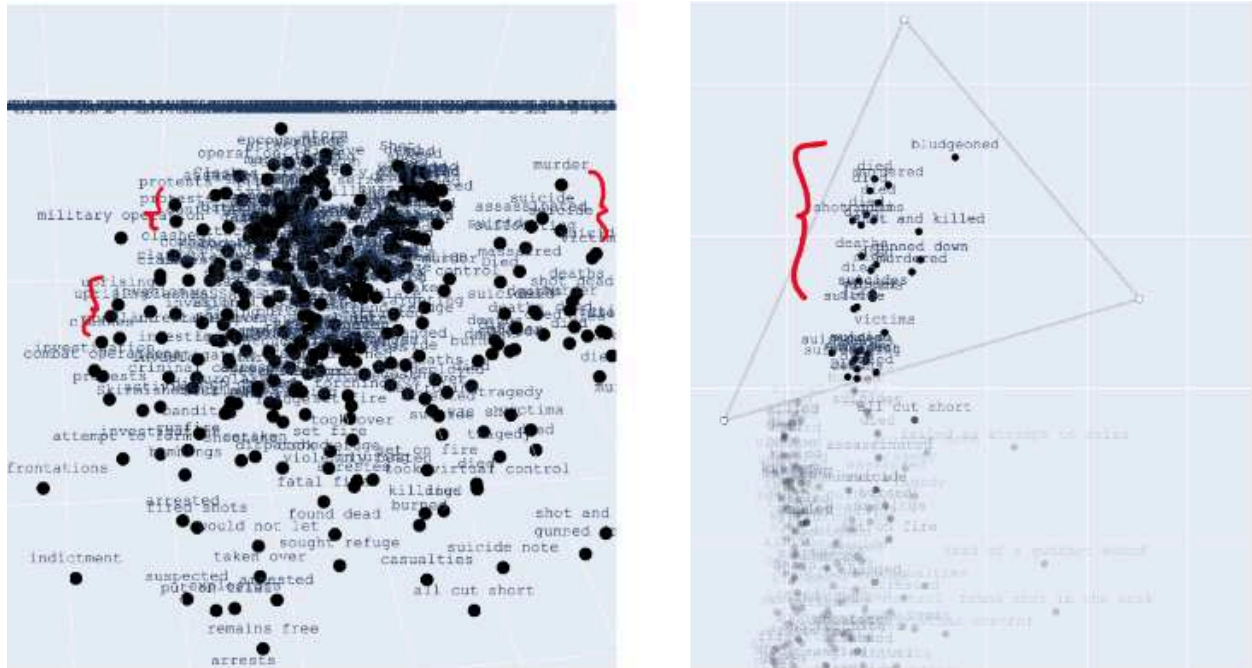


Figure 4.6: Principal Component Analysis (PCA) of Event embeddings generated from the default CDLM model. The left figure is a snapshot of the 3-D PCA while the right figure is a 2-D PCA snapshot. Red markers point to the visible event clusters.

Similarly, Fig. 4.7 shows that entity mentions describing the place of the event as well as the entity involved like "Kiev", "Oleg Kalashnikov" and "the entrance to his apartment block" are clustered in a relatively distinct space in the 3-PCA projection figure on the left. While organizational entities like "Federal Bureau of Investigation", "Central Intelligence Agency" and "foreign Intel agencies" are forming a distinct cluster in the right figure. These distinct clusters both in 2-D and 3-D suggest that coreference specific semantic information is definitely present in some form in the default CDLM model embeddings. Whether or not they are amenable to a high-fidelity semantic

transfer can only be checked after carrying out the affine transformations and verifying the results. The section below talks about the affine-mapping procedure.

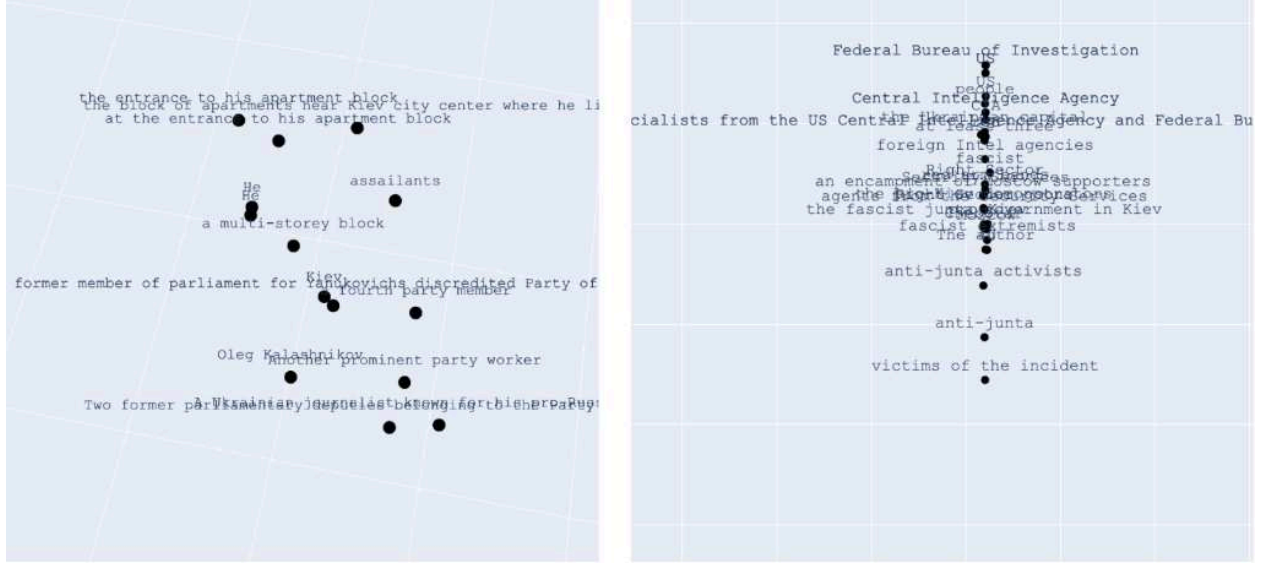


Figure 4.7: Principal Component Analysis of Entity embeddings generated from the default CDLM model. The left figure is a snapshot of the 3-D PCA while the right figure is a 2-D PCA snapshot.

4.5 Generating Affine Maps

In order to affine-map the source model embeddings to the target model spaces, we map the source and target space with a cartesian product combination. This leads to 12 Source→Target model spaces after eliminating the identical maps i.e., from each model to its own. Therefore, we do not map the latter four combinations and compare them in their native state. The affine mapping procedure we apply here is identical to the one we do for cognate-detection albeit a little simpler since we do not map words or sentences separately and instead, each sample is now a document containing a mention. In short, if V_s^i and V_t^j represent the **source** and the **target** embedding spaces, we carry out a least-square-based optimization method or ridge-regression [Hoerl and Kennard, 1970] between those spaces where i and j represent individual sample documents in

our LDC dataset for events and entities separately and assuming $\lambda=1$. β represents the L^2 -norm regularization parameter. Mathematically,

$$\text{Minimize}((V_t^j - V_s^i\beta)^T(V_t^j - V_s^i\beta) + \lambda\beta^T\beta),$$

4.6 Pairwise Clustering and Thresholding

In order to link the coreferring mentions for events as well as entities separately, we treat the entire clustering process separately for the two mention types. This clustering and scoring of paired mentions is carried out for each of the 12 affine-mapped embedding sets as well as the 4 native embedding sets. Fig. 4.5 provides a detailed step-by-step algorithm of our CDCR system with affine-mapping.

We first create sorted integer indices for all the paired mentions for each mention type. The number of paired combinations is significantly less than n -squared where n is the number of samples i.e., events or entities within the set of documents. This is because we ignore the identical combinations as well as the commutative alternative pair.⁴⁷ Using those sorted pairwise indices as rows and columns respectively, we create a square similarity matrix where each i (rows) and j (columns) represent the cosine similarity between that specific pair, with value of 1 across the main diagonal. Thereafter we create a sparse⁴⁸ adjacency matrix from the similarity matrix using the `scipy.sparse csr_matrix` package⁴⁹ using a chosen threshold T i.e., the adjacency matrix will only have positive links (or values of 1) for those pairs whose cosine similarity is greater than the chosen threshold. The non-adjacent vertices would be zeroes.

We experiment with three increasing cosine similarity thresholds i.e., 0.6, 0.75 and 0.95, for events as well as entities. The systems clusters and the labels are generated by finding the con-

⁴⁷If A and B are two event or entity mentions or their respective embeddings, we only consider $A \rightarrow B$ for calculating the cosine similarity while ignoring $B \rightarrow A$ as well as $A \rightarrow A$ or $B \rightarrow B$. We do this because the cosine metric is *symmetric* and to reduce *computational costs* of huge array-based multiplications.

⁴⁸Sparse matrices are mostly filled with zeroes and therefore, reduce computational or storage-related costs and are more amenable to matrix or vector operations.

⁴⁹https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csr_matrix.html

nected components of the adjacency matrix for each of those thresholds while the gold clusters and labels are already provided from the source LDC annotations. Thereafter, we use the CONLL perl-script scorer [Pradhan et al., 2014]⁵⁰ to score our systems clustering labels to our gold input. This gives us Precision, Recall and F1 scores for standard coreference metrics like BCUB and MUC.⁵¹ See section 4.7 below for a brief idea about coreference resolution evaluation metrics like BCUB and MUC.

4.7 Coreference Evaluation Metrics

In coreference resolution, the research community has used various metrics to evaluate results like the MUC, BCUB, CEAF as well as recent entity-linking-based metrics like the LEA [Moosavi and Strube, 2016]. While all the metrics mentioned above calculate recall, precision, and F1 scores to evaluate results, each of these metrics uses a different criterion to generate such scores. In general, the recall is an indicator⁵² of the fraction of correct coreference information, i.e., coreference links or entities, that is resolved in the systems output. In other words, it is the percentage of elements in a true reference chain (gold annotations) that are in the hypothesized reference chain (systems output). Similarly, precision is an indicator of the fraction of resolved coreference information that is correct. In other words, it is the percentage of elements in a systems cluster output that is already in the gold reference chain. And lastly, F1 score is the weighted harmonic mean of recall and precision.

More specific to this thesis is the MUC and the BCUB metric. MUC is one of the earliest systematic coreference evaluation metrics that was introduced by [Vilain et al., 1995]. It is a link-based metric as it computes recall based on the minimum number of *missing* links in the response entities in comparison to the key entities. It is also the least discriminative since it does

⁵⁰<https://github.com/conll/reference-coreference-scorers>

⁵¹We do not consider other metrics like CEAF or BLANC for this experiment. We leave them for future experiments.

⁵²In coreference resolution as commonly seen, both recall and precision are size-weighted to the respective cluster sizes and then averaged to get their final values. Since a coreference-resolution dataset usually contains more than one cluster, the overall recall, and precision scores act more as *indicators* instead of conclusive scores.

not differentiate whether an extra link merges two singletons. On the other hand, BCUB [Bagga and Baldwin, 1998] is a mention-based metric where the overall recall/precision is computed based on the recall/precision of the individual mentions. This makes it more sensitive to individual mentions and if such mentions happen to be singletons, such sensitivity disproportionately affects the evaluation scores. In our case, the gold annotation of the LDC Phase 1 dataset contains around 20 percent singletons for both events and entities, and since MUC is comparatively less sensitive to the presence of singletons than BCUB [Kübler and Zhekova, 2011] we have chosen the MUC metric as our primary evaluation metric.

4.8 Results and Analysis

As a reminder to the reader, our goal of doing CDCR using native and affine-mapped embeddings is not intended to reach a state-of-the-art performance but to study the semantic transfer phenomenon. The LDC Dataset used for evaluation is an on-going annotation process and is not publicly available for performance comparison with other peer-reviewed work. This thesis only attempts to answer questions regarding the nature and the extent of semantic encoding and transfer between the chosen embedding spaces and their affine-mapped spaces. This analysis only reveals the geometric vector space related properties of the language model embedding spaces that we have selected for this experiment. Other language models depending on the task may reveal something very different. The basis for model and baseline selection is how contextualized a model is, its relative embedding dimensions and computational feasibility. Moreover, for the purpose of this analysis, we have only considered the MUC coreference scores instead of BCUB since our gold annotations consist of at least 20 percent singleton mentions that affect BCUB performance disproportionately while MUC is relatively less affected by the presence of singletons [Kübler and Zhekova, 2011].

Native Model Performance: Encoding From an overall performance angle and as shown in Table 4.2 (events) and Table 4.3 (entities), the default native CDLM model performs the best for

Model Variant	Cosine Threshold = 0.6						Cosine Threshold = 0.75						Cosine Threshold = 0.95					
	BCUB			MUC			BCUB			MUC			BCUB			MUC		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CDLM	6.91	99.64	12.92	67.89	99.47	80.7	6.91	99.64	12.92	67.89	99.47	80.7	6.91	99.64	12.92	67.89	99.47	80.7
B_B	52.01	55.84	53.85	69.19	53.81	60.54	86.42	36.09	50.92	65.12	22.04	32.94	97.63	33.26	49.62	76	9.89	17.51
B_C	56.6	52.38	54.41	69.95	49.3	57.84*	78.61	39.47	52.55	70.54	31.59	43.64	97.63	33.28	49.64	75.34	9.54	16.94
B_L	65.07	47.06	54.62	70.2	42.53	52.97	91.46	35	50.63	72.53	17.88	28.69	97.63	33.26	49.62	76	9.89	17.51
B_B->B_C	54.24	54.53	54.38	70.46	52.6	60.23	50.4	50.06	50.23	60.31	47.22	52.96	83.18	32.46	46.69	12.57	3.47	5.44
B_B->B_L	61.03	50.39	55.2	72.1	47.56	57.32	58.75	40.33	47.83	45.06	29.34	35.54	83.63	32.46	46.77	13.46	3.64	5.73
B_B->CDLM	48.98	58.5	53.32	69.57	56.77	62.52	37.06	64.64	47.11	66.31	65.62	65.96	37.91	64.15	47.66	66.42	64.93	65.67
B_C->B_B	56.51	81.88	54.1	68.96	48.61	57.02	63.29	37.31	46.94	39.05	22.91	28.88	84.49	32.45	46.89	12.92	3.29	5.25
B_C->B_L	66.36	46.8	54.89	72.39	40.97	52.32	64.92	37.7	47.7	41.37	22.91	29.49	84.73	32.45	46.93	13.1	3.29	5.27
B_C->CDLM	53.64	53.86	53.75	68.77	50.86	58.48	42.93	58.06	49.36	64.2	57.29	60.55	43.68	57.31	49.58	64.1	56.42	60.01
B_L->B_B	51.56	56.5	53.92	69.55	54.34	61.01	62.23	34.6	44.47	29.79	18.05	22.48	83.07	32.46	46.68	12.5	3.47	5.43
B_L->B_C	55.38	53.81	54.54	70.54	51.56	59.57	53.27	46.69	49.76	57.54	42.36	48.8	82.62	32.46	46.6	12.26	3.47	5.41
B_L->CDLM	48.98	58.5	53.32	69.57	56.77	62.52	37.59	64.1	47.39	66.37	65.1	65.73	37.68	63.95	47.42	66.31	64.93	65.61
CDLM->B_B	42.29	58.92	49.24	65.08	61.8	63.4	35.52	59.39	44.46	59.49	60.93	60.2	78.35	32.62	46.06	13.52	4.86	7.15
CDLM->B_C	44.37	55.71	49.4	64.21	59.2	61.6	31.69	60.15	41.51	58.29	62.84	60.48	77.53	32.62	45.92	13.08	4.86	7.08
CDLM->B_L	52.4	51.85	52.12	65.66	53.12	58.73	34.59	58.47	43.47	57.92	60.24	59.06	78	32.62	46	13.33	4.86	7.12

Table 4.2: Final BCUB and MUC Scores i.e., Precision (P), Recall (R) and F1-scores (F1) for native and mapped embeddings on **LDC events**. CDLM, BERT_BASE (B_B), BERT_LARGE (B_L), BERT_COREF (B_C) are native models. Their affine-mapped variants are shown as Source→Target in the Model Variants column.

all three sub-metrics i.e., F-score, Precision and Recall within the MUC metric for both events and entities. For example for events, its F-scores, Precision and Recall are 80.7, 67.89 and 99.47 respectively. This is expected since this is the most "contextualized" model with CDCR task-specific pretraining along with additional finetuning using the special attention tokens (`<m>` and `</m>`). Moreover, this model can support much longer documents (4096 at a maximum) whereas the rest all have a maximum token length of 512. The native BERT-variants perform relatively lower⁵³ than native CDLM for smaller cosine thresholds but drastically lower for larger ones. For instance for events, native BERT-variants get F scores ranging from around 52 to 60 for lower thresholds while their F scores for higher thresholds are in the range of 17-18.

Among themselves, native BERT-variants perform almost equally well within respective cosine thresholds while overall losing performance with a higher cosine threshold. Assuming that an efficient CDCR system that does pairwise scoring and clustering requires semantic encoding of mentions within documents, the above performance trends suggests that such encoding does, in fact, occur on average. But, at least for *native* embedding spaces, its extent is not directly correlated with how "contextualized" or "large" the native model is. For instance, native BERT_LARGE, though larger with more encoder layers, consistently perform lower on average (F scores around 15 percent lower) than the other BERT-variants at lower cosine thresholds. For higher cosine thresholds of 0.95, though, this trend is less pronounced.

Affine-mapped Model Performance: Transfer In the case of exploring semantic transfer through affine-maps there arise cases where we see drastic enhancement in performance between native and mapped model spaces as well as between sets with opposite directionality of mapping (Source→Target). The idea is, if semantic transfer with a high-fidelity does occur after affine-mapping, it should lead to improved or at least equivalent performance for models like the BERT-variants post affine-mapping when compared to their native state. To put it differently, since we know that the CDLM is the most contextualized model and also the best-performing, can affine

⁵³All the performance comparisons in this paragraph and the later ones are based on F1 scores only.

Model Variant	Cosine Threshold = 0.6						Cosine Threshold = 0.75						Cosine Threshold = 0.95					
	BCUB			MUC			BCUB			MUC			BCUB			MUC		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CDLM	2.27	93.43	4.44	67.12	98.68	79.89	2.27	93.43	4.44	67.12	98.68	79.89	2.27	93.43	4.44	67.12	98.68	79.89
B_B	48.79	56.27	52.26	68.16	52.79	59.5	65.41	47.55	55.07	68.83	43.41	53.24	97.61	40.24	56.99	90.76	25.15	39.39
B_C	53.95	52.48	53.2	67.18	46.89	55.23	61.47	49.37	54.76	69.88	43.72	53.79	98.39	39.57	56.44	93.28	23.13	37.08
B_L	50.9	55.27	53	68.55	51.62	58.89	71.75	56.4	46.46	72.1	39.88	51.36	98.95	40.2	57.18	95.85	25.03	39.7
B_B->B_C	53.29	53.57	53.43	68.67	50.03	57.89	61.46	50.87	55.67	72.46	46.82	56.88	98.23	40.29	57.14	93.43	25.38	39.92
B_B->B_L	51.96	54.3	53.1	68.4	50.19	57.9	71.09	46.52	56.24	71.17	38.37	49.86	98.96	40.22	57.2	95.83	24.96	39.6
B_B->CDLM	48.77	55.8	52.05	67.56	52.32	58.97	48.82	55.8	52.08	67.63	52.32	59	49.37	55.56	52.28	67.89	52.05	58.92
B_C->B_B	55.81	51.56	53.6	67.28	45.27	54.12	70.84	46.08	55.84	70.37	36.82	48.34	99.02	39.4	56.37	95.7	22.44	36.35
B_C->B_L	52.29	49.49	53.95	67.11	42.71	52.2	76.44	44.24	56.05	72.08	33.52	45.76	99.06	39.35	56.33	95.99	22.8	36.17
B_C->CDLM	53.79	52.21	52.99	66.61	46.55	54.8	53.82	52.21	53	66.64	46.55	54.81	54.44	51.92	53.15	66.68	46.16	54.55
B_L->B_B	49.37	55.89	53.92	68.3	52.28	59.23	68.71	46.75	55.64	70.2	40.54	51.4	98.33	40.23	57.1	93.63	25.07	39.55
B_L->B_C	54.03	53.45	53.74	68.88	49.76	57.78	62.97	50.02	55.75	72.59	45.58	56	98.44	40.28	57.17	94.37	25.34	39.96
B_L->CDLM	48.71	56.08	52.13	67.81	52.59	59.24	48.86	55.92	52.16	67.75	52.44	59.12	49.22	55.81	52.31	67.89	52.28	59.07
CDLM->B_B	23.57	70.34	35.31	66.58	81.93	73.46	39.46	62.24	48.3	68.02	71	69.48	93.1	46.92	62.39	86.44	43.25	57.65
CDLM->B_C	20.63	71.05	31.97	66.01	83.64	73.79	36.58	63.88	46.52	68.04	73.95	70.87	91.53	46.99	62.1	84.51	44.22	58.06
CDLM->B_L	23.77	69.62	35.44	66.12	81.47	72.99	40.95	61.02	49.01	67.74	70	68.85	93.15	46.88	62.37	86.5	43.21	57.63

Table 4.3: Final BCUB and MUC Scores i.e., Precision(P), Recall(R) and F1-scores(F1) for native and mapped embeddings on **LDC entities**. CDLM, BERT_BASE (B_B), BERT_LARGE (B_L), BERT_COREF (B_C) are native models. Their affine-mapped variants are shown as Source→Target in the Model Variants column.

maps to and from CDLM lead to any performance difference, preferably improvement? How does the answer to this question change between events and entities and what does it tell us about the vector space related geometric characteristics of the chosen models?

Our observations from Table 4.2 and Table 4.3 answer some of the questions above. For example, CDLM in its native state not only tops performance in all three sub-metrics but also remains *constant* across all thresholds for both events and entities. This implies, and quite conclusively so, that more contextualized language models like CDLM encode its semantic information *anisotropically* [Ethayarajh, 2019a] i.e., within a narrow cone on its 768 dimensional space. This is surprising because until now, we saw such anisotropy having a positive correlation with only smaller-sized models in the cognate-detection section. It seems this is true for more "contextualized" models too.

Moreover, for events as seen in Table 4.2, the yellow-highlighted cells in red show relatively improved performance for a cosine threshold of 0.6 but drastic improvement for larger thresholds when BERT-variants (source) are affine-mapped to the native CDLM (target) compared to native BERT-variants. For instance, BERT-variants like BERT_LARGE and Coref_BERT when affine-mapped to the native CDLM (target) show F scores of 62.52 and 58.48 respectively for a threshold of 0.6 while their native versions showed F scores of 52.97 and 57.84 respectively. Similarly, for BERT_LARGE and Coref_BERT and for a higher threshold of 0.95, such scores improve drastically to 65.61 and 60.01 while their native model scores were 17.51 and 16.94 respectively. This implies that less contextualized models like the BERT-variants can benefit from semantic information transfer from more contextualized models when the latter is the target embedding space.

Interestingly, this is true **regardless** of the directionality (Source→Target) of the affine mapping but its extent depends on the cosine threshold. For instance, for events (the non-highlighted cells in red in Table 4.2) and more so for entities (the non-highlighted cells in red in Table 4.3), we see relatively improved performance for a cosine threshold of 0.6 and drastic improvement for larger thresholds of 0.75 for CDLM (source) affine-mapped to the BERT-variants (target) when

compared to native BERT-variants. More specifically for events, in this case of reversed directionality, BERT-variants like BERT_LARGE and Coref_BERT affine-mapped from the native CDLM (source) show improved F scores of 58.73 and 61.6 respectively for a lower threshold of 0.6 while their native versions showed F scores of 52.97 and 57.84 respectively. Similarly, for BERT_LARGE and Coref_BERT and for a higher threshold of 0.75, such scores improve drastically to 59.06 and 60.48 while their native model scores were 28.69 and 43.64 respectively. However, for an even larger cosine threshold of 0.95, we see a drastic decrease in performance specific to *events* (but not for entities) where F scores reduce to almost single digits (around 7) for all BERT-variants. We discuss this specific trend below in the paragraph on event-specific difficulties in affine-transfer for BERT-variants.

Another surprising observation is BERT_BASE affine-mapped to BERT_LARGE space outperforming native BERT_LARGE for events for a threshold of 0.6. This is marked in the non-highlighted teal colored cell in Table 4.2 for events. For instance, BERT_BASE affine-mapped to BERT_LARGE shows an F score of 57.32 while native BERT_LARGE gets an F score of 52.97. Similarly, for entities, BERT_LARGE affine-mapped to BERT_BASE space outperforms native BERT_LARGE as shown in the non-highlighted teal colored cell in Table 4.3. This is despite the implicit noisiness in a transformation that goes up in dimensionality since BERT_LARGE embeddings are 1024 dimensional. This and the above trends suggest that semantic transfer in language models can lead to even improved results depending on the cosine threshold compared to native models. This is a surprising result when compared vision-specific works [McNeely-White, 2020, McNeely-White et al., 2022].

And now to take a dig at BERT! It appears that BERT-variants have a more difficult time resolving event mentions than entities for higher cosine thresholds. For instance, in Table 4.2, the non-highlighted red colored cells show significantly reduced scores for a high cosine threshold of 0.95 for CDLM (source) mapped to BERT-variants (target) when compared to native CDLM. The scores are actually in single digits for events (around 7) for all three BERT-variants while the scores for entities for the same threshold and affine-mapping direction are around 58 for all three

BERT-variants. This trend is important because it is unique to events and we do not see such drastic reduction (in fact, we see an improvement) in case of entities. This implies that BERT-variants encode less *event-specific* semantic information than entity-related even though they are less-contextualized than CDLM on average. Events are more difficult to resolve within CDCR in general [Humphreys et al., 1997] and our observations with affine mappings are consistent with such previous results. Perhaps, the corpora used to train such large language models do not have enough event-specific information on average compared to entities but this is beyond the scope of this thesis to explore. Another conjecture could be that BERT-variants encode event-specific semantic information more anisotropically than that of entities. However, answering this question would require further geometric analysis of the event or entity-specific embedding spaces. One way would be to check if this trend is repeated for other affine transformations like shear and rotation maps.

Chapter 5

CONCLUSION AND FUTURE WORK

Our observations from both the tasks suggest that a *canonical* embedding space, that is amenable to semantic transfer through affine-mapping, perhaps exists for language models as apparently is the case for image embeddings [McNeely-White et al., 2022]. That all Transformer-based language models (the ones we selected, in total nine) encode and transfer semantics albeit to a varying extent **despite** architectural variations and native-model task-agnosticity (since native models were not specifically trained for our two tasks) buttresses the argument for the existence of this space. A conclusive comment on its existence surely warrants more exploration on the intricacies of how such embedding spaces are "learned" in the pretraining phase. Another way to verify its existence would be to see how well they generalize to other downstream NLP tasks. Exploring the veracity of its existence in the case of intermediate encoder layers of such language models is relevant here too.

Nevertheless, our observations suggest that while it is true that semantic information is transferred by linear maps on average, the extent to which that happens is model-specific. In other words, semantic information transfer depends on the source and the target model space. There is also a strong argument to be made here that the embedding spaces of the native models and their affine maps are perhaps **task-agnostic** too, apart from being canonical, since they **generalize** well. In our case, the models we have selected differ in their training architectures, the number of parameters in their weight matrices, hyperparameters selected as well the linguistic variety of their respective corpora. Perhaps, Transformer-based bidirectional encoder systems that rely on the self-attention mechanism encode the semantics of a language that is transferable because it is task-agnostic. The latter characteristic is interesting because this is unlike what previous work [McNeely-White et al., 2020a, McNeely-White et al., 2022] had found. Their model sample space contained CNN models trained specifically for image classification or facial recognition while their evaluation of linear maps was on the *same* or a similar task. In our case, our model sam-

ple space is more varied. For instance, the large language models we had selected were trained for masked-word prediction and next-sentence prediction primarily, but their embedding-based affine maps were evaluated on a significantly different tasks like cognate detection and CDCR. Perhaps, the training regime is what makes such semantic information transferable with high consistency regardless of the task (as long as the task requires some semantic information for resolution).

In case of the cognate detection task, we observed cases of semantic encoding that robustly transferred (through affine-mapping) language-specific semantics helpful for cross-lingual tasks. We also learnt that deep neural networks that have non-linear decision boundaries are our best bet to effectively use such transferred information in disambiguating negative samples to achieve better results. Our observations also reveal interesting vector space related geometric properties of the embedding spaces. That is, larger models are more isotropic and therefore, more responsive to affine-mapped semantic transfer to smaller monolingual model spaces. For CDCR, we found that affine-maps deployed on coreference-specific language models encode semantic information in distinct spaces depending on how contextualized a model space is. For instance, we observed that a more contextualized model like native CDLM encodes its coreference-specific semantic information anisotropically i.e., in a tight high dimensional cone while the BERT-variants, whether native or affine-mapped, does so in a more dispersed manner. Perhaps, the extent of isotropy is *not* directly correlated with such semantic transfer since it could be task-specific but it definitely needs more exploration.

More specifically for CDCR, we also saw the strange case where BERT_BASE affine-mapped to BERT_LARGE space outperforming native BERT_LARGE for events despite dimensionality-expansion related perturbations. This could solve an important but often ignored problem in deploying language models like BERT, the one of computational cost-benefits. If linearly transformed embeddings create spaces that outperform those of native variants, one can visualize ways to neurally align embeddings generated from less-contextualized models at the fine tuning stage *without* additional pretraining. Our final observation that BERT variants are iffy about resolving event mentions makes us wonder if the problem lies in how representative of events their corpora is.

It could also be that we just need richer contextual representations for a quintessentially difficult task. Perhaps, this warrants an exploration of intermediate layer embeddings for a task-specific richer representation of embeddings.

The above observations pique our curiosity about a few interesting ideas for future work. A natural extension of this work would be to use affine-mapping in the case of other Transformer-based language models like the autoregressive or generative models like GPT-2 [Radford et al., 2019] since we have only used encoder-only language models for this work. This will answer the question whether a canonical space exists for all Transformer-based language models or just the masked-word-based representational ones like BERT. Another potential direction could be exploring the usefulness of semantic transfer for classically syntactic NLP tasks like span boundary prediction or next-sentence prediction. For task-specific future work, exploring the extent of cross-lingual semantic transfer for low-resource languages (LRLs) could surely help improve representation for such languages in the AI/NLP community. In CDCR, since we have only used MUC for the evaluation, one could try other metrics like LEA or BLANC that are more fair to the coreference-specific idiosyncrasies of the LDC Dataset. While cosine similarity metrics do help determine how similar two documents (or two mentions) are for CDCR, a better way to use such language models would be to get more "contextualized" pairwise representations straight out of the model before any clustering steps using a cross-encoder architecture [Cattan et al., 2021a]. Lastly, since affine maps are minimally constrained, other more constrained linear transformations like shear or a rotation maps may expose more interesting results about the vector spaces of language models.

Bibliography

- [Ammer, 2013] Ammer, C. (2013). *The American Heritage dictionary of idioms: American English idiomatic expressions & phrases*. HMH.
- [Artetxe et al., 2018a] Artetxe, M., Labaka, G., and Agirre, E. (2018a). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- [Artetxe et al., 2018b] Artetxe, M., Labaka, G., and Agirre, E. (2018b). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.
- [Bagga and Baldwin, 1998] Bagga, A. and Baldwin, B. (1998). Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.
- [Bagga and Baldwin, 1999] Bagga, A. and Baldwin, B. (1999). Cross-document event coreference: Annotations, experiments, and observations. In *Coreference and Its Applications*.
- [Barhom et al., 2019] Barhom, S., Shwartz, V., Eirew, A., Bugert, M., Reimers, N., and Dagan, I. (2019). Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4179–4189.
- [Bejan and Harabagiu, 2014] Bejan, C. A. and Harabagiu, S. (2014). Unsupervised event coreference resolution. *Computational Linguistics*, 40(2):311–347.
- [Beltagy et al., 2020] Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv e-prints*, pages arXiv–2004.

- [Bharadwaj et al., 2016] Bharadwaj, A., Mortensen, D. R., Dyer, C., and Carbonell, J. G. (2016). Phonologically aware neural model for named entity recognition in low resource transfer settings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1472.
- [Björkelund and Kuhn, 2014] Björkelund, A. and Kuhn, J. (2014). Learning structured perceptions for coreference resolution with latent antecedents and non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 47–57.
- [Bloodgood and Strauss, 2017] Bloodgood, M. and Strauss, B. (2017). Using global constraints and reranking to improve cognates detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1983–1992.
- [Brown et al., 2008] Brown, C. H., Holman, E. W., Wichmann, S., and Velupillai, V. (2008). Automated classification of the world’s languages: a description of the method and preliminary results. *Language Typology and Universals*, 61(4):285–308.
- [Caciularu et al., 2021] Caciularu, A., Cohan, A., Beltagy, I., Peters, M. E., Cattan, A., and Dagan, I. (2021). Cdln: Cross-document language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2648–2662.
- [Cattan et al., 2021a] Cattan, A., Eirew, A., Stanovsky, G., Joshi, M., and Dagan, I. (2021a). Cross-document coreference resolution over predicted mentions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5100–5107.
- [Cattan et al., 2021b] Cattan, A., Eirew, A., Stanovsky, G., Joshi, M., and Dagan, I. (2021b). Realistic evaluation principles for cross-document coreference resolution. In *Proceedings of* SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 143–151.

- [Chen and Ji, 2009] Chen, Z. and Ji, H. (2009). Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 54–57, Suntec, Singapore. Association for Computational Linguistics.
- [Church and Kordoni, 2022] Church, K. W. and Kordoni, V. (2022). Emerging trends: Sota-chasing. *Natural Language Engineering*, 28(2):249269.
- [Cieri et al., 2020] Cieri, C., Fiumara, J., Strassel, S., Wright, J., DiPersio, D., and Liberman, M. (2020). A progress report on activities at the linguistic data consortium benefitting the lrec community. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3449–3456.
- [Clark and Manning, 2016] Clark, K. and Manning, C. D. (2016). Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.
- [Conneau et al., 2020] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, É., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.
- [Conneau and Lample, 2019] Conneau, A. and Lample, G. (2019). *Cross-Lingual Language Model Pretraining*. Curran Associates Inc., Red Hook, NY, USA.
- [Crystal, 2011] Crystal, D. (2011). *A dictionary of linguistics and phonetics*. John Wiley & Sons.
- [Dellert, 2018] Dellert, J. (2018). Combining information-weighted sequence alignment and sound correspondence models for improved cognate detection. In *Proceedings of the 27th international conference on computational linguistics*, pages 3123–3133.
- [Denis and Baldridge, 2007] Denis, P. and Baldridge, J. (2007). Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 236–243.

- [Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- [Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. N. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.
- [Durrett and Klein, 2013] Durrett, G. and Klein, D. (2013). Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1971–1982.
- [Ethayarajh, 2019a] Ethayarajh, K. (2019a). How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65.
- [Ethayarajh, 2019b] Ethayarajh, K. (2019b). Rotate king to get queen: Word relationships as orthogonal transformations in embedding space. *arXiv preprint arXiv:1909.00504*.
- [Everitt and Skrondal, 2010] Everitt, B. S. and Skrondal, A. (2010). The cambridge dictionary of statistics.
- [Ganesan et al., 2021] Ganesan, A., Ferraro, F., and Oates, T. (2021). Learning a Reversible Embedding Mapping using Bi-Directional Manifold Alignment. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3132–3139.
- [Haddow and Kirefu, 2020] Haddow, B. and Kirefu, F. (2020). Pmindia—a collection of parallel corpora of languages of india. *arXiv preprint arXiv:2001.09907*.

- [He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [Hoerl and Kennard, 1970] Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.
- [Hu et al., 2020] Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., and Johnson, M. (2020). Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.
- [Humphreys et al., 1997] Humphreys, K., Gaizauskas, R., and Azzam, S. (1997). Event coreference for information extraction. In *Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*.
- [Jäger, 2018] Jäger, G. (2018). Global-scale phylogenetic linguistic inference from lexical resources. *Scientific Data*, 5(1):1–16.
- [Jawahar et al., 2019a] Jawahar, G., Sagot, B., and Seddah, D. (2019a). What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.
- [Jawahar et al., 2019b] Jawahar, G., Sagot, B., and Seddah, D. (2019b). What does bert learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657.
- [Jurafsky, 2000] Jurafsky, D. (2000). *Speech & language processing*. Pearson Education India.

- [Jurafsky and Martin, 2009] Jurafsky, D. and Martin, J. H. (2009). Speech and language processing.
- [Kakwani et al., 2020] Kakwani, D., Kunchukuttan, A., Golla, S., N.C., G., Bhattacharyya, A., Khapra, M. M., and Kumar, P. (2020). IndicNLP Suite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.
- [Kanojia et al., 2021a] Kanojia, D., Patel, K., Bhattacharyya, P., Kulkarni, M., and Haffari, G. (2021a). Utilizing wordnets for cognate detection among indian languages. *arXiv preprint arXiv:2112.15124*.
- [Kanojia et al., 2021b] Kanojia, D., Sharma, P., Ghodekar, S., Bhattacharyya, P., Haffari, G., and Kulkarni, M. (2021b). Cognition-aware cognate detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3281–3292.
- [Kenyon-Dean et al., 2018] Kenyon-Dean, K., Cheung, J. C. K., and Precup, D. (2018). Resolving event coreference with supervised representation learning and clustering-oriented regularization. *arXiv preprint arXiv:1805.10985*.
- [Khanuja et al., 2021] Khanuja, S., Bansal, D., Mehtani, S., Khosla, S., Dey, A., Gopalan, B., Margam, D. K., Aggarwal, P., Nagipogu, R. T., Dave, S., et al. (2021). Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.
- [Kondrak, 2001] Kondrak, G. (2001). Identifying cognates by phonetic and semantic similarity. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.
- [Kübler and Zhekova, 2011] Kübler, S. and Zhekova, D. (2011). Singletons and coreference resolution evaluation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 261–267.

- [Kudo and Richardson, 2018] Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- [Labat and Lefever, 2019] Labat, S. and Lefever, E. (2019). A classification-based approach to cognate detection combining orthographic and semantic similarity information. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 602–610, Varna, Bulgaria. INCOMA Ltd.
- [Lee et al., 2012] Lee, H., Recasens, M., Chang, A., Surdeanu, M., and Jurafsky, D. (2012). Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500.
- [Lefever et al., 2020] Lefever, E., Labat, S., and Singh, P. (2020). Identifying cognates in English-Dutch and French-Dutch by means of orthographic information and cross-lingual word embeddings. In *PROCEEDINGS OF THE 12TH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC 2020)*, pages 4096–4101. European Language Resources Association (ELRA).
- [Lenc and Vedaldi, 2015] Lenc, K. and Vedaldi, A. (2015). Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 991–999.
- [Levenshtein et al., 1966] Levenshtein, V. I. et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.
- [List, 2014] List, J.-M. (2014). Investigating the impact of sample size on cognate detection. *Journal of Language Relationship*, 11(1):91–102.

- [Markov, 1954] Markov, A. A. (1954). The theory of algorithms. *Trudy Matematicheskogo Instituta imeni VA Steklova*, 42:3–375.
- [McNeely-White et al., 2020a] McNeely-White, D., Beveridge, J. R., and Draper, B. A. (2020a). Inception and resnet features are (almost) equivalent. *Cognitive Systems Research*, 59:312–318.
- [McNeely-White et al., 2020b] McNeely-White, D., Sattelberg, B., Blanchard, N., and Beveridge, R. (2020b). Exploring the interchangeability of cnn embedding spaces. *arXiv preprint arXiv:2010.02323*.
- [McNeely-White et al., 2022] McNeely-White, D., Sattelberg, B., Blanchard, N., and Beveridge, R. (2022). Canonical face embeddings. *IEEE Transactions on Biometrics, Behavior, and Identity Science*.
- [McNeely-White, 2020] McNeely-White, D. G. (2020). *SAME DATA, SAME FEATURES: MODERN IMAGENET-TRAINED CONVOLUTIONAL NEURAL NETWORKS LEARN THE SAME THING*. PhD thesis, Colorado State University.
- [Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- [Miller, 1995] Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- [Moosavi and Strube, 2016] Moosavi, N. S. and Strube, M. (2016). Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642.
- [Mortensen et al., 2018] Mortensen, D. R., Dalmia, S., and Littell, P. (2018). Epitran: Precision g2p for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

- [Mortensen et al., 2016] Mortensen, D. R., Littell, P., Bharadwaj, A., Goyal, K., Dyer, C., and Levin, L. (2016). Panphon: A resource for mapping ipa segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484.
- [Mulloni and Pekar, 2006] Mulloni, A. and Pekar, V. (2006). Automatic detection of orthographics cues for cognate recognition. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC06)*.
- [Nath et al., 2022] Nath, A., Ghosh, R., and Krishnaswamy, N. (2022). Phonetic, Semantic, and Articulatory Features in Assamese-Bengali Cognate Detection. In *Workshop on NLP for Similar Languages, Varieties, and Dialects (Vardial)*. ACL.
- [Ng and Cardie, 2002] Ng, V. and Cardie, C. (2002). Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 104–111.
- [Ogueji et al., 2021] Ogueji, K., Zhu, Y., and Lin, J. (2021). Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126.
- [Palmer et al., 2005] Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.
- [Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- [Pires et al., 2019] Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

- [Pradhan et al., 2014] Pradhan, S., Luo, X., Recasens, M., Hovy, E., Ng, V., and Strube, M. (2014). Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.
- [Pustejovsky et al., 2003] Pustejovsky, J., Castano, J. M., Ingria, R., Sauri, R., Gaizauskas, R. J., Setzer, A., Katz, G., and Radev, D. R. (2003). Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.
- [Radford et al., 2019] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- [Rahman and Ng, 2011] Rahman, A. and Ng, V. (2011). Coreference resolution with world knowledge. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 814–824.
- [Rama and List, 2019] Rama, T. and List, J.-M. (2019). An automated framework for fast cognate detection and bayesian phylogenetic inference in computational historical linguistics. Association for Computational Linguistics.
- [Rama et al., 2018] Rama, T., List, J.-M., Wahle, J., and Jäger, G. (2018). Are automatic methods for cognate detection good enough for phylogenetic reconstruction in historical linguistics? *arXiv preprint arXiv:1804.05416*.
- [Rijhwani et al., 2019] Rijhwani, S., Xie, J., Neubig, G., and Carbonell, J. (2019). Zero-shot neural transfer for cross-lingual entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6924–6931.
- [Salton, 1989] Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., USA.

- [Schafer and Yarowsky, 2002] Schafer, C. and Yarowsky, D. (2002). Inducing translation lexicons via diverse similarity measures and bridge languages. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.
- [Sennrich et al., 2016] Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- [Smith et al., 2017] Smith, S. L., Turban, D. H. P., Hamblin, S., and Hammerla, N. Y. (2017). Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv e-prints*, page arXiv:1702.03859.
- [Soon et al., 2001] Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544.
- [Suárez et al., 2019] Suárez, P. J. O., Sagot, B., and Romary, L. (2019). Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache.
- [Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.
- [Tenney et al., 2019] Tenney, I., Das, D., and Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline. *arXiv e-prints*, pages arXiv–1905.
- [Turton et al., 2021] Turton, J., Smith, R. E., and Vinson, D. (2021). Deriving contextualised semantic features from bert (and other transformer model) embeddings. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 248–262.

- [Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [Verleysen and François, 2005] Verleysen, M. and François, D. (2005). The curse of dimensionality in data mining and time series prediction. In *International work-conference on artificial neural networks*, pages 758–770. Springer.
- [Vilain et al., 1995] Vilain, M., Burger, J. D., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.
- [vor der Brück and Pouly, 2019] vor der Brück, T. and Pouly, M. (2019). Text similarity estimation based on word embeddings and matrix norms for targeted marketing. 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 1827–1836.
- [Yang et al., 2015] Yang, B., Cardie, C., and Frazier, P. (2015). A hierarchical distance-dependent bayesian model for event coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:517–528.
- [Ye et al., 2020] Ye, D., Lin, Y., Du, J., Liu, Z., Li, P., Sun, M., and Liu, Z. (2020). Coreferential reasoning learning for language representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186.
- [Yu et al., 2022] Yu, X., Yin, W., and Roth, D. (2022). Pairwise representation learning for event coreference. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 69–78.

- [Zeng et al., 2020] Zeng, Y., Jin, X., Guan, S., Guo, J., and Cheng, X. (2020). Event coreference resolution with their paraphrases and argument-aware embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3084–3094.
- [Zhukova et al., 2021] Zhukova, A., Hamborg, F., Donnay, K., and Gipp, B. (2021). Xcoref: Cross-document coreference resolution in the wild. *CoRR*, abs/2109.05252.