

ABSTRACT

Title of proposal: CLOSING THE USER EXPECTATIONS
GAP: INTERPRETABILITY
AND CULTURAL EXTENSIONS
TO NLP MODELS

Fenfei Guo, 2024

Dissertation directed by: Professor Jordan Boyd-Graber
Department of Computer Science
College of Information Studies
Language Science Center
Institute for Advanced Computer Studies

Data-driven approaches, particularly large pre-trained language models (LLMs), have revolutionized natural language processing (NLP) by effectively capturing semantics and memorizing complex structures from extensive unstructured text data, resulting in substantial improvements in NLP applications. However, data-driven models might not always reflect user needs and expectations due to several inherent limitations and gaps, such as bias in training data, lack of interpretability and transparency, over-reliance on common patterns, etc.

This proposal addresses several aspects to close the gap between data-driven NLP models and user expectations, with a focus on improving interpretability and extending the models' ability to align with different cultural aspects.

Interpretability is vital to many use cases. For example, in the context of sense representation learning, users may need to interpret the semantics of each representa-

tion to build sense inventories or study the chronological shifts of word meaning. In this proposal, we both improve the interpretability of the sense model and evaluation metrics. Specifically, we design human-centric tasks to evaluate the model’s ability to learn interpretable distinct sense inventories and examine how well the learned senses align with human judgments. To reveal the human-distinguishable sense structures for word embeddings, we propose a modified Gumbel Softmax function to mimic hard sense selection. Without word sense induction data, the model automatically learns to distinguish senses based on context. Our model learns human-distinguishable sense inventories without reducing performance on computer-centric semantic evaluations. Improving the interpretability of the learned representations provides more understandable and diverse usage of the learned representations for users.

Besides interpretability, gaps often exist between the training domain and user domains. In real applications, aligning the generated content with user expectations is crucial. In this proposal, we draw attention to cultural aspects that have been overlooked and aim to extend the data-driven model’s capabilities to address these use cases.

Firstly, singable song translation is a practical necessity in cultural communications, such as the overseas production of Disney movies and musical theater. However, this is a hard task even for human experts, since the translated lyrics must match the prosody of the preexisting music in addition to retaining the original meaning. To meet the user needs in singable song translation, where in-domain data is scarce, we incorporate human expert rules as priors to control the lyrics

generation process, producing singable and intelligible translations without supervised data. We address the overlooked yet vital aspect of aligning pitch in tonal languages’ lyrics with melody, significantly affecting singability and intelligibility. Based on musicians’ and linguists’ expertise, we summarize three criteria impacting these aspects and guide the decoding phase’s beam search with designed constraints. Ultimately, we create the first automatic Mandarin song translation system without parallel song translation data, generating singable and intelligible Mandarin lyric translations based on human evaluations.

Further, in the proposed work, we address the cultural differences that have been overlooked in cross-cultural communications and data collection. Specifically, machine translation aims to facilitate communication between people who speak different languages. However, people from different cultures may not share the same context to understand each other. Existing models focus on direct language translation while overlooking cultural differences. We propose an adaptive machine translation (adaptive MT) approach to address this issue. Our approach translates the original source text while adapting the culturally specific entities in the original sentence, along with the context, into their counterparts in the target culture. In practice, besides direct usage in communication, adaptive MT can help resolve data bias in developing multilingual models and QA systems or build connections in cross-cultural document analysis, among other applications.

CLOSING THE USER EXPECTATIONS GAP:
INTERPRETABILITY AND CULTURAL EXTENSIONS TO NLP
MODELS

by

Fenfei Guo

Dissertation proposal submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2024

Advisory Committee:
Professor Jordan Boyd-Graber, Advisor
Professor 1
Professor 2
Professor 3

© Copyright by
Fenfei Guo
2024

Table of Contents

List of Tables	v
List of Figures	vi
1 Introduction	1
1.1 Organization	2
1.1.1 Background	2
1.1.2 Improving the interpretability of learned representation	2
1.1.3 Introducing cultural specific constraints with inductive bias from human knowledge	3
1.1.4 Adaptive machine translation for cross-cultural communication with RAG	4
1.1.5 Conclusion	4
2 Background	5
2.1 Data-driven NLP Models	5
2.1.1 Learning from data	5
2.1.2 The role of data-driven models in NLP	6
2.1.3 Human prior knowledge and data-driven NLP models	8
2.2 Distributed Representation Learning	8
2.2.1 Vector semantics and sparse vector representation	9
2.2.2 Word2vec	9
2.3 Pretrained Large Language Model	11
2.3.1 Pretraining and Finetuning	11
2.3.2 Self-Attention Networks: Transformers	12
2.3.3 Transformers as language models	14
2.3.4 Cross-lingual Language Models	16
3 Uncover Interpretable Senses from Words	18
3.1 The necessity of human interpretable sense representations	18
3.2 Attentional Sense Induction	19
3.2.1 Foundation: Gumbel Softmax	19

3.2.2	Reveal Word Senses in Skip-Gram Model	20
3.2.2.1	Why Attention? Musing on Alternatives	20
3.2.2.2	Attentional Sense Induction	20
3.2.2.3	Scaled Gumbel Softmax for Sense Disambiguation	21
3.3	Evaluating Interpretability	22
3.3.1	Word Intrusion for Sense Coherence	24
3.3.2	Contextual Word Sense Selection	25
3.4	Word Similarity Evaluation	26
3.4.1	Word Similarity vs. Interpretability	28
3.5	Related Work: Representation, Evaluation	29
3.5.1	Granularity	30
3.6	Conclusion	31
4	Human-knowledge Guided Automatic Song Translation for Tonal Languages	32
4.1	Introduction	32
4.2	Background: Prose, Poetry, and Song Translation	34
4.2.1	Song Translation for Tonal Languages	35
4.2.2	Mandarin Tones and how to Sing them	35
4.3	AST for Tonal Languages	36
4.3.1	Criteria	36
4.3.2	Task Definition	37
4.3.3	Aligning Lyrics to Music	38
4.3.3.1	Length Alignment	38
4.3.3.2	Pitch Alignment	38
4.3.3.3	Rhythmic Alignment with Word Segmentation in Mandarin	39
4.4	GagaST	40
4.4.1	Song-Text Style Translation	40
4.4.2	Length Control	41
4.4.3	Music Guided Alignment Constraints	41
4.5	Experiments	41
4.5.1	Training Datasets and Model Configuration	41
4.5.2	Evaluation Dataset	42
4.5.3	Evaluation Metrics	42
4.5.3.1	Objective Evaluation	42
4.5.3.2	Subjective Evaluation	43
4.5.4	Hyper-parameters and Trade-offs	43
4.5.5	Evaluation Results	44
4.5.5.1	Objective Evaluation Results	44
4.5.5.2	Subjective Evaluation Results	45
4.6	Related Work	46
4.6.1	Verse Generation and Translation	46
4.6.2	Constrained Text Generation	46
4.6.3	Lyrics Generation	46
4.7	Conclusion	46

5	Proposed Work	48
5.1	Adaptive Machine Translation	48
5.1.1	Cross-cultural Entity Adaptions	48
5.1.2	Complete Cross-cultural Contexts	48
5.1.3	Over Generation and Fact Verification Model	49
5.1.4	Applications and Analysis	49
6	Conclusion and Timeline	50
6.1	Timeline	50
A	Reading List	52
A.1	Representation Learning	52
A.2	Pretrained Language Model	53
A.3	Human Knowledge Controlled Generation	54

List of Tables

3.1	Word intrusion evaluations on top ten nearest neighbors of sense embeddings. Users find misfit words most easily with GASI- β , suggesting these representations are more interpretable.	23
3.2	Human-model consistency on <i>contextual word sense selection</i> ; P is the average probability assigned by the model to the human choices. GASI- β is most consistent with crowdworkers. Reducing sense duplication by retraining our model with pruning mask improves human-model agreement.	24
3.3	Similarities of human and model choices when they agree and disagree for two metrics: simple word overlap (top) and Glove cosine similarity (bottom). Humans agree with the model when the senses are distinct.	25
3.4	A case where MSSG has low overlap but confuses raters (agreement 0.33); the model chooses s1.	25
3.5	Spearman’s correlation 100ρ on SCWS (trained on 1B token, 300d vectors except for Huang et al.). GASI and GASI- β both can disambiguate the sense and correlate with human ratings. Retraining the model with pruned senses further improves local similarity correlation.	28
3.6	Spearman’s correlation on non-contextual word similarity (MaxSim). GASI- β has higher correlation on three datasets and is competitive on the others. PFT-GM is trained with two components/senses while other models learn three. A full version including MSSG is in appendix.	29
4.1	A piece of song “Seasons of love” from the musical <i>Rent</i> . We convert the notes into a normalized numerical pitch level for actual computation.	37
4.2	Objective results on test set of GagaST with different constraints under one-to-one and one-to-many assignments. All results here use the same pre-training checkpoint and length tags are applied. For length score, 9 (0.09) means that 9 out of 713 samples are longer than the predefined length with an average ratio 0.09.	43
4.3	Subjective evaluation results for GagaST w/o constraints and GagaST.	45
6.1	Adaptation MT paper timeline	51

List of Figures

2.1	The Transformer architecture (Vaswani et al., 2017)	13
2.2	Overall pre-training and fine-tuning procedures for BERT (Devlin et al., 2019). [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).	14
2.3	On the left: BERT replaces random tokens with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently. On the right: GPT predicts tokens auto-regressively and can be used for generation. However, words can only condition on leftward context, so it cannot learn bidirectional interactions.	15
2.4	The inputs of BART does not need to be aligned with decoder outputs, allowing arbitrary noise transformations. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an auto-regressive decoder.	16
2.5	The pretraining framework of Multilingual BART (left), where two corruption methods are used (1) sentence permutation (2) word-span masking as the injected noise; and fine-tuning on downstream MT tasks (right). A special language id token is added at both the encoder and decoder.	17
3.1	Network structure with an example of our GASI model which learns a set of global context embeddings \mathbf{C} and a set of sense embeddings \mathbf{S} .	19
3.2	t-SNE projections of nearest neighbors for “bond” by <i>hard-attention</i> models: MUSE (RL-based) and our GASI- β . Trained on same dataset and vocabulary, both models learn three vectors per word (bond_ i is i^{th} sense vector). GASI (right) learns three distinct senses of “bond” while MUSE (left) learns overlapping senses.	23
4.1	Example Mandarin translations for “Let it go” in <i>Frozen</i> . Of these, only the official human song translation considers whether a singer could sing the song: it fits the length of the notes and matches the tones with the pitch of notes.	33

4.2	In total languages like Mandarin, the pitch changes the meaning of the words (left). Each of the four tones in Mandarin (right) has a different pitch profile. Figure from Xu (1997).	34
4.3	A misheard example in Mandarin song caused by a mismatch between music pitch flow and the lyric’s tones. The heard word is “sǐ zài” instead of “sì zài”, because notes are going up and “sì zài” is going down by the sandhi of Mandarin tone.	35
4.4	The alignments of syllables in Mandarin to notes in the song “Love Island”. Orange: <i>REST</i> notes; Blue: cases where one syllable is assigned to a group of multiple notes (need consider <i>tone shape</i> alignment, e.g., the down arrow matches with falling tone of “ràng”); Green: cases where one syllable is assigned with one note.	36
4.5	For the translated songs in Mandarin to be singable, the transition directions of successive music pitch should align with that of the tones of successive characters. The arrows show the acceptable transition directions (summarized in this paper) in music for two successive Mandarin characters (w_{i-1}, w_i) based on the shape of Mandarin tones and the sandhi of tones.	37
4.6	Overview of GagaST for English–Mandarin song translation. We first pre-train a lyrics translation model with mixture domain data (left); and then add alignment constraints in decoding scoring function during inference (right), we use unconstrained version as our baseline in the experiment.	40
4.7	Trade-off between semantics and lyric-music alignments; all curves are drawn for the valid set.	44

Chapter 1: Introduction

As a subdomain of artificial intelligence, natural language processing ([Jurafsky and Martin, 2000](#), NLP) develops algorithms that enable computers to process human language so that it performs tasks involving human language, such as analyzing texts, enabling human-machine communication, improving human-human communication or helping human to create contents.

Over the past fifty years, the development of natural language processing has progressed from state machines and rule-based systems to statistical models and neural models. These models are increasingly data-driven, relying less on human-designed rules and more on machine learning algorithms trained on large datasets. In recent years, self-supervised data-driven models have significantly advanced the field of natural language processing. Unlike supervised models which learn from human-annotated text data or parallel data like human-translated transcripts, self-supervised models learn from large amounts of unlabeled raw texts by predicting masked contexts given part of the original texts. These models learn from vast amounts of unstructured text and capture patterns and structures present in the data. These patterns are then generalized to various tasks through parameter sharing, enabling models to achieve state-of-the-art performance across a diverse array of applications, such as low-resource machine translation ([Lewis et al., 2020](#)), sentiment analysis ([Xu et al., 2019](#)), and question-answering tasks ([Karpukhin et al., 2020](#)).

However, such models have their limitations as well. First of all, the pattern abstractions learned are implicit and hidden within non-interpretable model parameters. Therefore, while these models have captured an immense amount of knowledge, the means to access that knowledge remain limited, non-interpretable, and inefficient. Without interpretable intermediate steps, the reliability of the outputs is difficult to verify, and if a prediction is incorrect, it becomes challenging to identify the root cause and make necessary improvements, as the reasoning behind the model’s behavior is hidden within its complex structures. While for many applications, such as healthcare and finance, transparency and reliability are crucial. Moreover, the potential ways for human-AI interactions would also be limited if the model is non-interpretable, as users are unable to understand the rationale behind the model-generated results, making it difficult to engage with intermediate steps or provide accurate controls. Even if the primary concern is the accuracy of outputs without interpretability, fine-tuning and applying large language models is computationally expensive and time-consuming. This inefficiency can be particularly problematic in real-time or resource-constrained applications, where rapid and

efficient computation is essential, such as streaming translation or applications on portable devices with poor internet connections.

Another major limitation of data-driven models is that they heavily rely on data to learn patterns and may struggle to generalize effectively if the data distribution is imbalanced. Even extremely large pre-trained language models like OpenAI’s GPT models—which are trained on over 300 billion tokens and comprise more than 175 billion parameters—still exhibit varying performance across domains with differing amounts of data (Brown et al., 2020; Ouyang et al., 2022; OpenAI*, 2023). For example, if we query ChatGPT with GPT4 to provide a Mandarin word that has the same tone pattern as 电话 (diànhuà; phone), it provides 音乐 (yīnyuè; music), which is incorrect since 电话 (diànhuà; phone) has pattern as “4th-tone, 4th-tone” while 音乐 (yīnyuè; music) has “1st-tone, 4th-tone”. However, if we query directly what tones the two words “电话” and “音乐” have, it provides the correct answer. The model has sufficient knowledge to draw the analogy but fails to do so, presumably, such queries are not present in their training data. Humans, on the other hand, can easily draw a correct analogy if they possess the relevant knowledge, as they have the ability to make generalizations based on rules rather than relying on memorizing data patterns.

To summarize, data-driven models have brought huge gains in various NLP tasks, however, their lack of interpretability and inability to generalize based on rules have limited their applicability in various domains. In this thesis, we improve the interpretability of data-driven models by explicitly revealing their hidden structures, introduce new evaluation metrics to specifically examine the model’s interpretability; and we incorporate human knowledge as an inductive bias into these models to provide control over model outputs. We extend the ability of data-driven NLP models in different cultural aspects.

1.1 Organization

1.1.1 Background

We propose to improve the data-driven NLP models with human knowledge. Chapter 2 provides a brief overview of the development of data-driven models in natural language processing (NLP). We introduce representation learning and pre-trained language models, discussing the benefits of using large amounts of unlabeled data. We then briefly address the importance of incorporating human knowledge and making models more amenable to integrating such knowledge. Finally, we present the word embedding model and the transformer-based language model that will be employed in this thesis.

1.1.2 Improving the interpretability of learned representation

Word representation learning models, such as word2vec (Mikolov et al., 2013a), have brought significant improvements to downstream tasks. However, these models do not consider that words also have multiple senses; for example, the English word

"bank" can refer to a financial institution that accepts deposits from customers and makes loans, or the side of a river. If users need to interact at the sense level, such as creating a sense inventory or examining changes in word usage over time, they must put in extra effort to clustering the learned word representations given contexts or collecting data to train a classifier. Thus, a primary goal for multi-sense word embeddings is to learn human understandable sense features. Unfortunately, existing multi-sense models have only been evaluated on computer-centric dimensions and have ignored the question of sense interpretability.

Chapter 3 exposes the discrepancy between computer-centric tasks such as word similarity tasks and human interpretability. We propose a new coherence evaluation for sense embeddings that examines how well the learned sense inventory by sense embeddings align with human judgments. And we present a simple model to explicitly reveal word senses and learn sense-specific embeddings. Compared to existing sense-specific embedding models, our focus is on ensuring the interpretability of the learned sense embeddings via differentiable hard sense selection. Our model learns human distinguishable sense inventories without reducing performance on computer-centric evaluations.

1.1.3 Introducing cultural specific constraints with inductive bias from human knowledge

Pretrained language models have gained success in various language generation tasks, including question answering, translation, and creative writing. However, in real applications, aligning the generated content with user expectations is crucial. For example, to compose compelling poems, the content must adhere to user-expected formats and themes. In other cases, such as translating song lyrics, the output must also follow specific musical rules. However, parallel song translation data for fine-tuning is scarce. In these scenarios, we propose to incorporate inductive bias from human knowledge into data-driven models and provide interpretable controls on generated content.

Chapter 4 draws attention to a long-overlooked factor—the alignment between the tones of lyrics and melody—which affects significantly the singing and listening experiences for audiences who speak tonal languages. Without training data for song lyrics translation for tonal languages, we incorporate human priors into the decoders to generate lyrics that align with the music. We review literature from musicians and linguists, introduce two often overlooked yet essential qualities of songs in automatic systems—singability and intelligibility—that relate to human experiences, and summarize rules based on expertise that improve these two qualities for tonal languages. Building on these rules, we develop both automatic and human evaluation methods and design constraints that we incorporate into our model. Ultimately, we create the first automatic song translation system for Mandarin without any parallel song translation data, generating Mandarin song lyric translations that are both singable and intelligible based on human evaluations.

1.1.4 Adaptive machine translation for cross-cultural communication with RAG

Representation models and pre-trained language models have captured enormous world knowledge from the vast amount of text data they read. Retrieving knowledge from these pre-trained models can help to enrich the context knowledge in many applications.

For example, in cross-cultural communication, direct translation is not enough, since people from different cultural will lack mutual background context to understand each other. For instance, if a student from the United States talks about sports and mentions football athletes to a Chinese student, the Chinese student may have difficulties in grasping the essence of what the U.S. student is trying to say, even if all the terms are correctly translated, since he/she knows nothing about American football.

Chapter 5 proposes a novel task—cross-cultural adaptive machine translations (adaptive MT)—to help remove friction in cross-cultural communications. That is, we do not just perform direct translation, but also adapt the entity of interest (EoI) in the original sentence, along with the context, into its counterparts in the target culture. In practice, besides direct usage in communication, adaptive MT can also help resolve data bias in developing multilingual models and QA systems, or build connections in cross-cultural document analysis, etc.

1.1.5 Conclusion

Finally, in Chapter 6, we conclude this proposal and gives a timeline for the remaining research work until defense.

Chapter 2: Background

This chapter reviews the development of data-driven models in natural language processing (NLP). We introduce representation learning and pretrained language models, discussing the benefits of using large amounts of unlabeled data. We then briefly address the importance of incorporating human knowledge and making models more amenable to integrating such knowledge. Finally, we present the word embedding model and the transformer-based language model that will be employed in this thesis.

2.1 Data-driven NLP Models

This section introduces the definition of data-driven models and reviews the development of data-driven models in NLP.

2.1.1 Learning from data

Data-driven models refer to models that learn patterns and relationships directly from data, rather than relying on handcrafted rules or expert knowledge. The models learn from data (what we already know) to generalize to new cases (what we don't know) and make predictions. There are two main types of learning tasks with different goals. **Supervised learning** methods learn from labeled data with known pairings between input X_{seen} and output Y_{seen} with the goal of predicting unseen outputs Y_{unseen} by estimating the probability of Y given X , $P(Y|X)$. **Un-supervised learning**, on the other hand, refers to cases where the labeled data is unavailable and the model learns from unlabeled data X_{seen} , discovers latent factors Z within the data that can generalize to new cases X_{unseen} by estimating $P(X, Z)$. Recently, **self-supervised learning** has gained success in various domains such as computer vision, audio processing, and natural language processing where the raw input data present intrinsic latent structures. Same to unsupervised learning, self-supervised learning handles cases where the “labels” are not provided. Instead, it learns to predict part of the input data from other parts, $P(X_i|X_j)$. Therefore, self-supervised learning can be viewed as a subcategory of unsupervised learning where the supervisory signal is derived from the input data itself.

2.1.2 The role of data-driven models in NLP

NLP enables computers to process human language and encompasses a wide range of tasks that involve making predictions or generating outputs based on text inputs, such as machine translation, sentiment analysis, named entity recognition, question answering, etc.

Before data-driven models, most NLP systems are rule-based. As one of the most compact means to exchange knowledge naturally, human language is highly abstract and exhibits structured and regular patterns. These structural rules govern the composition of words, phrases, and clauses in a language. Consequently, traditional approaches that process language are rule-based and rely on expert knowledge to manually craft rules and heuristics. For example, one of the earliest and most well-known chatbot systems, ELIZA (Weizenbaum, 1966), uses pattern matching and substitution based on regular expressions (Kleene et al., 1956) to simulate conversations with humans. However, these approaches are limited in their ability to capture the nuances and variations of natural language and often require extensive manual labor to maintain and update.

On the other hand, data-driven models learn patterns directly from data, which typically consists of large amounts of text. Compared to rule-based models, data-driven models are more adept at handling nuances and variations present in the data, and they can generalize better to unseen texts. Early data-driven NLP models use **statistical methods** to model language. Examples include N-gram models, Hidden Markov Models (Rabiner, 1989), Topic Models (Blei et al., 2003), Statistical Machine Translation models (Brown et al., 1990; Koehn et al., 2003), etc. These models capture language patterns by estimating pattern probabilities based on counting frequencies and then perform inferences based on probabilistic rules. With the advancement of computational resources and increased access to vast amounts of raw text data, **neural network models** start to gain success in NLP (Bengio et al., 2003) and soon dominate the field with an ever-growing capability to learn from enormous amounts of text data.

Section 2.1.1 introduces the concept of unsupervised learning, in which the model captures the intrinsic patterns present in unlabeled data X_{seen} and generalizes to new cases X_{unseen} . **Unsupervised learning** methods are particularly useful in NLP, because language exhibits intrinsic syntactic and semantic structures. Unsupervised models can leverage vast amounts of unlabeled raw text and capture these structures. There are two essential types of unsupervised NLP models: representation learning models and language models.

Topic models discover hidden semantic structures or topics within a collection of documents. Latent Dirichlet Allocation (Blei et al., 2003, LDA) is the most popular topic modeling algorithm that assumes each document in a corpus is a mixture of various topics, and each topic is a distribution of words. LDA analyzes the co-occurrence patterns of words in documents, estimating the underlying topic distributions for both documents and words, thus providing a means to organize, explore, and summarize large text collections.

Representation learning models learn vector representations of the data

that make it easier to extract useful information when building classifiers or other predictors (Bengio et al., 2013). In NLP, representation learning models learn vector representations for various levels of language units, including words, phrases, sentences, documents, etc. Initially, researchers create one-hot features with statistical methods, such as N-grams, bag-of-words (Joachims, 2005), and term frequency-inverse document frequency (Salton and Buckley, 1988, TF-IDF). These features capture word occurrences and co-occurrences in texts. However, these features are sparse and suffer from the curse of dimensionality, limiting their ability to generalize to unseen word sequences. With the development of neural network models, **distributed representation learning**, such as word embeddings like Word2Vec (Mikolov et al., 2013a,b) and GloVe (Pennington et al., 2014), further revolutionize feature representation in NLP. These models learn continuous dense vector representations from raw texts by predicting one word from another within a small context window,

$$P(w_{\text{context}}^i | w_{\text{pivot}}). \quad (2.1)$$

The learned representation captures semantic relationships between words in the encoding space, i.e., the distance between these representations can reflect semantic similarities.

Language models, on the other hand, captures not only semantics but also syntactic structures. They assign probabilities to sequences of words or tokens in a text. They aim to predict the next word in a sequence given the preceding words,

$$P(w_t | w_{t-1}, \dots, w_0), \quad (2.2)$$

or to estimate the probability of a given sequence of words,

$$P(w_t, \dots, w_0). \quad (2.3)$$

N-gram language model counts the frequency of n-gram, which is a contiguous sequence of n words from a given text, and estimates the probability of each word sequence (Equation 2.3) with sequence frequency. While neural language models estimate word sequence probability with chain rule,

$$P(w_T, \dots, w_0) = \prod_{t=0}^T P(w_t | w_{t-1}, \dots, w_0), \quad (2.4)$$

and the model learns to optimize the next-word prediction probability (Equation 2.2). Neural network language models, especially pretrained Large Language Models (LMs), significantly enhance the capabilities of language models. Their deep structures excel at capturing complex syntactic and semantic structures and contextual knowledge present in language from vast amounts of text data. By transferring the captured patterns and knowledge from pretrained language models to downstream tasks through fine-tuning, we can achieve significantly better performance using less labeled data, which is often difficult to obtain. In Section 2.3, we introduce pretrained LMs with more details and present several models that will be used in this proposal.

Without labeled data, both distributed representation learning models and language models learn to predict part of the data from other parts (Equation 2.1 and 2.2). Therefore, these models are also referred to as **self-supervised models**.

2.1.3 Human prior knowledge and data-driven NLP models

As NLP model development progresses, there is an increasing reliance on data rather than on human prior knowledge. From distributional representation learning to pretrained language models, leveraging large unlabeled text corpus and capturing complex language patterns with increasingly deeper and larger neural networks enables NLP models to better generalize to unseen data. Transferring knowledge from these models to downstream applications helps achieve state-of-the-art results in a wide range of NLP tasks, such as Machine Translation for low-resource languages (Gu et al., 2018; Liu et al., 2020) and Question Answering tasks (Karpukhin et al., 2020; Sachan et al., 2022).

However, human remains the center of NLP tasks. Many tasks require interpreting and communicating results to end-users, especially the content generation tasks such as chatbots, lyrics generation, etc. Even for extremely large pretrained language model, GPT3 with 175 million parameters (Brown et al., 2020), existing work shows that learning from human feedback through reinforcement learning (Ouyang et al., 2022) significantly improves the truthfulness of the generated outputs by LLM and reduces the toxic outputs. On the other hand, humans leverage the rich contextual and world knowledge captured by learned representations or pretrained LMMS to perform tasks that require such knowledge, such as studying diachronic drift of words (Hamilton et al., 2016a), finding cross-cultural entity adaptations (Peskov et al., 2021), or perform document analysis (Kusner et al., 2015), etc.

However, unlike statistical methods such as HMMs and topic models, neural network models are substantially less interpretable. This is because the learned abstractions are implicit, hidden within their non-interpretable deep structures. Consequently, while these models have succeeded in capturing an immense amount of knowledge, the means to access or interact with that knowledge remain limited, non-interpretable, and inefficient. In contrast, it is more straightforward to incorporate human knowledge into statistical methods as probabilistic priors (Hu et al., 2011). Moreover, data-driven models heavily rely on data to learn patterns, while in many cases, the data distribution is imbalanced. Even extremely large pre-trained language models like OpenAI’s GPT models—which are trained on over 300 billion tokens and comprise more than 175 billion parameters—still exhibit varying performance across domains with differing amounts of data (Brown et al., 2020; Ouyang et al., 2022; OpenAI*, 2023).

In summary, while data-driven NLP models, particularly those based on neural networks, have revolutionized the development of NLP, the integration of human prior knowledge into such models remains crucial. In this thesis, our focus is on the construction of more interpretable representation learning models and the incorporation of human priors as inductive biases into neural generation models.

2.2 Distributed Representation Learning

As we explain in Section 2.1.1, distributed representation learning derives continuous dense vector representations for words, phrases, or sentences directly from

raw texts. Finding self-supervised ways to learn representations of the input, instead of creating representations by hand via feature engineering is an important focus of NLP research (Bengio et al., 2013). The learned dense vector representations are called embeddings. They capture the semantic meaning and syntactic relations of language. We present the most widely used word embedding model Word2vec (Mikolov et al., 2013a,b) in the following subsection. Based on this model, we reveal the hidden sense structures of words and induce interpretable sense-specific embeddings in Chapter 3.

2.2.1 Vector semantics and sparse vector representation

The idea of using vectors to represent word semantics originates from works dating back to the 1950s. Osgood et al. (1957) use a point in three-dimensional space to represent the connotation of a word. Meanwhile, linguists (Joos, 1950; Harris, 1954; Firth, 1957) define the meaning of a word by its distribution in language use and form the **distributional hypothesis**, i.e., words that occur in similar contexts tend to have similar meanings.

Traditionally, the vector representations of words and terms are generally built based on co-occurrence matrix, i.e., a matrix where its element represents how often words co-occur in the corpus (*term-term matrix*) or their occurrence in documents (*term-document matrix*). More advanced methods such as TF-IDF and positive pointwise mutual information (Church and Hanks, 1989; Dagan et al., 1993; Niwa and Nitta, 1994, PPMI) use weighted probabilities instead of simple frequencies to represent words (co-)occurrence. However, the dimensionality of such vector representations grows linearly with the size of the vocabulary or the number of documents. This results in an exceedingly sparse and high-dimensional representation, leading to computational inefficiency and the curse of dimensionality. That is, machine learning models require a substantially larger sample size to achieve generalizability when the feature dimensionality is high. To reduce the inconvenience of sparse high-dimensional vector representations and diminish the curse of dimensionality, Deerwester et al. (1990) perform latent semantic analysis (LSA) and use matrix decomposition method, singular value decomposition (SVD), to create low-dimension representations of terms from the term-document matrix.

2.2.2 Word2vec

With the development of neural network models, Bengio et al. (2003) build a neural network language model (NNLM), which is a feedforward neural network that optimizes the probability of sequences of words (Equation 2.3) from input corpus, while each word is represented by a dense and continuous vector with a fixed dimension. Therefore, NNLM not only learns a statistical language model but also yields a set of distributed representations of words.

Following this work, Mikolov et al. (2013a) propose a new architecture, Word2vec, which efficiently learns continuous dense vector representations of words from large amounts of unlabeled text. Word2Vec models are built based on the distributional

hypothesis and learn word vectors by optimizing the probability of a word’s occurrence given its surrounding words. The learned word vectors are generally referred to as word embeddings.

Word2vec models have two variations: the continuous Skip-Gram model and the continuous Bag-of-Words (CBOW) model. Considering the following sentence,

...I really enjoy reading books on rainy days...

if we choose the word “books” as the pivot word and a context window size of three, i.e.,

...I [really enjoy reading **books** on rainy days]...

$c_1 \quad c_2 \quad c_3 \quad w \quad c_4 \quad c_5 \quad c_6$

the CBOW model maximizes the probability of “books” given its context words [*really, enjoy, reading, on, rainy, days*], $P(w|c_1, c_2, \dots, c_6)$; while the Skip-Gram model maximizes the probability of each context word given the pivot word, $P(c_i|w), i = 1, \dots, 6$.

Specifically, both models learn two sets of vectors embeddings, the word embeddings $\mathbf{W} \in \mathbb{R}^{|V| \times d}$ and the context embeddings $\mathbf{C} \in \mathbb{R}^{|V| \times d}$, where $|V|$ is the size of the vocabulary $|V|$ and d is the predefined vector dimension (50, 100, 300, etc). We use w_i to represent a certain pivot word and \mathbf{w}_i as its vector representation from \mathbf{W} ; c_j^i represents the j -th context word of w_i in the context window \tilde{c}_i , and \mathbf{c}_j^i as the vector representation from \mathbf{C} .

Skip-Gram model maximizes the likelihood of each context word c_j^i that surrounds a given pivot word w_i in a context window \tilde{c}_i ,

$$J_{SG}(\mathbf{W}, \mathbf{C}) \propto \sum_{w_i \in V} \sum_{c_j^i \in \tilde{c}_i} \log P(c_j^i | w_i; \mathbf{W}, \mathbf{C}). \quad (2.5)$$

To model the probability of a context word c_j^i given the pivot word w , Mikolov et al. (2013a) use the dot product $\mathbf{c}_j^i \top \mathbf{w}_i$ to estimate the similarity of the two vectors (and the words they represent), and compute the $P(c_j^i | w_i)$ with a softmax function over the whole vocabulary V ,

$$P(c_j^i | w_i; \mathbf{W}, \mathbf{C}) = \frac{\exp(\mathbf{c}_j^i \top \mathbf{w}_i)}{\sum_{c_k \in V} \exp(\mathbf{c}_k \top \mathbf{w}_i)}. \quad (2.6)$$

CBOW model maximizes the likelihood of the pivot word w_i given the context window \tilde{c}_i ,

$$J_{CBOW}(\mathbf{W}, \mathbf{C}) \propto \sum_{w_i \in V} \sum_{c_j^i \in \tilde{c}_i} \log P(w_i | \tilde{c}_i; \mathbf{W}, \mathbf{C}), \quad (2.7)$$

similarly, the CBOW model estimates the likelihood using the softmax function on the word context similarity over the vocabulary,

$$P(w_i | \tilde{c}_i; \mathbf{W}, \mathbf{C}) = \frac{\exp(\tilde{\mathbf{c}}_i \top \mathbf{w}_i)}{\sum_{w_k \in V} \exp(\tilde{\mathbf{c}}_i \top \mathbf{w}_k)}, \quad (2.8)$$

where $\bar{\mathbf{c}}_i$ is computed based on the bag-of-words assumption (Joachims, 2005),

$$\bar{\mathbf{c}}_i = \frac{1}{|\tilde{\mathbf{c}}_i|} \sum_{\mathbf{c}_j^i \in \tilde{\mathbf{c}}_i} \mathbf{c}_j^i. \quad (2.9)$$

Negative Sampling However, computing the softmax over the full vocabulary is extremely resource-consuming. Inspired by noise contrasting estimation (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012), Mikolov et al. (2013a) approximate the likelihood $\log P(\mathbf{c}_j^i | \mathbf{w}_i)$ with negative sampling. Taking the Skip-Gram model as an example, instead of predicting the context word \mathbf{c}_j^i , the new task is to distinguish \mathbf{c}_j^i from randomly sampled T negative samples \mathbf{v}_j^t . Formally, the updated Skip-Gram negative sampling objective for pivot word \mathbf{w}_i becomes,

$$\log \sigma(\mathbf{c}_j^{i\top} \mathbf{w}_i) + \sum_{t=1}^T \mathbb{E}_{\mathbf{v}_j^t \sim P_n(v)} \left[\log \sigma(-\mathbf{v}_j^{t\top} \mathbf{w}_i) \right], \quad (2.10)$$

where σ is the Sigmoid function,

$$\sigma(x) = \frac{1}{1 + e^{-x}},$$

and the noise sample \mathbf{v}_j^t is drawn from the noise distribution $P_n(v)$,

$$P_n(v) \propto U(v)^{3/4} \quad (2.11)$$

2.3 Pretrained Large Language Model

Word embedding models are shallow neural networks with only one layer (the embedding layer). They capture word semantics from surrounding contexts but not syntactic structures, which limits their applications in NLP. With the development of neural network models and computational resources, deeper and larger network structures have been successfully applied in NLP, such as Convolutional Neural Networks (LeCun et al., 1998; Kim, 2014, CNNs) and Long Short-Term Memory networks (Hochreiter and Schmidhuber, 1997; Sutskever et al., 2014, LSTM). Peters et al. (2018) pretrain a bi-directional LSTM with language modeling objectives and learn contextualized word embeddings with the internal states of the bi-directional language model. These internal states not only capture contextual information but also model aspects of syntax, which results in better performance in downstream NLP tasks than other embeddings.

2.3.1 Pretraining and Finetuning

Pretraining followed by finetuning is essentially a transfer learning technique, which aims to improve the learning of the target predictive function $f_T(\cdot)$ for the target task \mathcal{T}_T in target domain \mathcal{D}_T through transferring knowledge from a source

model that is trained on source domain \mathcal{D}_S for source task \mathcal{T}_S , where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$ (Pan and Yang, 2010). In general, transfer learning can be categorized into two settings:

1. In the *Inductive transfer learning* setting, the target task is different from the source task ($\mathcal{T}_T \neq \mathcal{T}_S$). In this case, some labeled data are required in the target domain \mathcal{D}_T to induce the predictive model $f_T(\cdot)$.
2. In the *Transductive transfer learning* setting, the target task and the source task are the same ($\mathcal{T}_T = \mathcal{T}_S$), but the target domain is different from the source domain ($\mathcal{D}_T \neq \mathcal{D}_S$), and labeled data is only available in the source domain \mathcal{D}_S .

Howard and Ruder (2018) adapt the idea of inductive transfer learning, i.e., directly from computer vision (Sharif Razavian et al., 2014) and introduce a Universal Language Model Fine-tuning methods (ULMFiT) for text classification. The idea of pretraining and subsequent direct fine-tuning on the whole learned language model architecture starts to gain success in NLP. Building upon the revolutionary new self-attention network architecture (Vaswani et al., 2017), the Transformer, the practice of pretraining larger and deeper language models with various objectives and fine-tuning on downstream tasks achieves state-of-the-art performance across a diverse array of applications in NLP. Compared to distributed representation learning, pretrained large language models capture not only rich semantics but also complex syntactic structures within their deep network structures. In this thesis, we further incorporate human knowledge as inductive biases into pretrained language models for generation tasks and improve its performance on zero-shot generation task. We introduce briefly the Transformer architecture in Section 2.3.2, and present in Section 2.3.3 various pretrained language models, including the models that we are going to use in this thesis.

2.3.2 Self-Attention Networks: Transformers

Sequence-to-sequence neural models with an encoder-decoder structure have gained success in language generation tasks, such as machine translation (Bahdanau et al., 2014; Cho et al., 2014; Sutskever et al., 2014) and question answering (Seo et al., 2017). These models are generally based on the recurrent neural network architecture LSTM. In which, the **encoder** maps an input sequence of text tokens $\mathbf{x} = (x_1, \dots, x_n)$ to a sequence of continuous representations $\mathbf{z} = (z_1, \dots, z_n)$. Given \mathbf{z} , the decoder then generates an output sequence of tokens $\mathbf{y} = (y_1, \dots, y_m)$ one element at a time. At each step the model is auto-regressive (Graves, 2013), consuming the previously generated token y_{t-1} as additional input when generating the next y_t . That is, the model optimizes the following likelihood of generating \mathbf{y} given \mathbf{x} ,

$$\log P(\mathbf{y}|\mathbf{x}) = \sum_{t=1}^m \log P(y_t|\mathbf{x}, y_1, \dots, y_{t-1}) = \sum_{t=1}^m \log P(y_t|\mathbf{z}, y_1, \dots, y_{t-1}) \quad (2.12)$$

where \mathbf{z} is the learned hidden representations from input \mathbf{x} by the encoder.

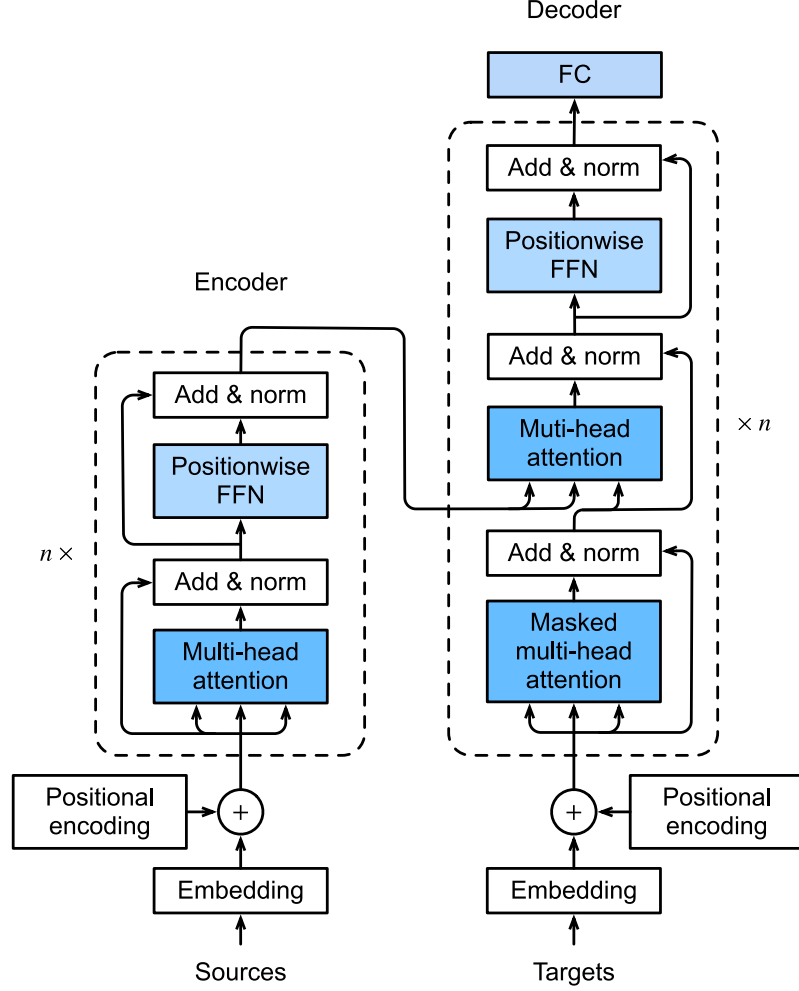


Figure 2.1: The Transformer architecture (Vaswani et al., 2017)

The Transformer follows this overall encoder-decoder architecture using stacked self-attention layers and point-wise fully connected layers for both the encoder and decoder (Figure 2.1). Specifically, the **encoder** is composed of a stack of N identical layers. Each layer has two sub-layers: the multi-head self-attention layer; and the point-wise fully connected feed-forward network layer. The output of each sub-layer is applied with a layer normalization (Ba et al., 2016) and a residual connection (He et al., 2016), i.e, $\mathbf{x}' = \text{LayerNorm}(\mathbf{x} + \text{Sublayer}(\mathbf{x}))$, where \mathbf{x}' is the final output of each sub-layer given \mathbf{x} . The **decoder** is also composed of a stack of N identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs **multi-head attention** over the output hidden representation \mathbf{z} of the encoder stack. The tokens (y_t, \dots, y_m) that haven't been decoded yet are masked in the first multi-head attention sub-layer.

The **multi-head attention** architecture is the key component of the Transformer architecture, which allows the model to jointly attend to information from different representation subspaces at different positions. Within this architecture,

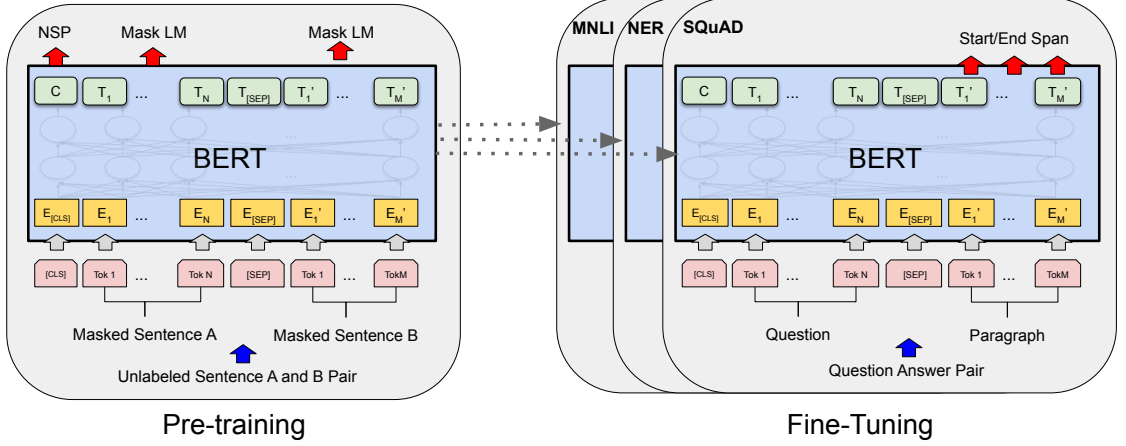


Figure 2.2: Overall pre-training and fine-tuning procedures for BERT (Devlin et al., 2019). [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

each **scaled dot-product attention** computes an attention distribution based on the input queries and keys of dimension d_k , and values of dimension d_v . In practice, the attention function is computed on a set of queries simultaneously, packed together into a matrix Q . The keys and values are also packed together into matrices K and V . The matrix outputs are computed as,

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V. \quad (2.13)$$

Instead of performing a single attention function with d_{model} -dimensional keys, values and queries, in each head, the **multi-head attention** linearly project the queries, keys and values h times with different learned linear projections to d_k , d_k and d_v dimensions, respectively. The final output of the multi-head attention is the average of the concatenation of the output of each head with a learned linear projection W^O ,

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O, \quad (2.14)$$

where each head is computed as,

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \quad (2.15)$$

and the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

2.3.3 Transformers as language models

The Transformer architecture, with its multi-head mechanism, has demonstrated strong capabilities in capturing semantics and learning complex syntactic structures. This has led it to achieve state-of-the-art performance in machine translation tasks. With the emergence of pretraining and finetuning techniques in NLP,



Figure 2.3: On the left: BERT replaces random tokens with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently. On the right: GPT predicts tokens auto-regressively and can be used for generation. However, words can only condition on leftward context, so it cannot learn bidirectional interactions.

researchers develop various pretrained language models based on this powerful Transformer architecture. There are two branches of pretrained language models: the ones that are optimized with prediction objectives such as the masked language model (MLM); and the generative pretraining models that optimize the likelihood of the generated outputs.

With prediction objectives BERT (Devlin et al., 2019) is the first and most popular transformer-based pretrained language model. It has a bidirectional encoder-decoder transformer architecture (Figure 2.2) and is pretrained with prediction loss objective: masked language modeling and next sentence prediction. Specifically, for masked language modeling, a subset of input tokens is masked and the model learns to predict masked tokens from the context; for next sentence prediction, the model learns to predict whether a pair of sentences are contiguous. The pre-training data consists of the BooksCorpus and English Wikipedia. BERT achieves state-of-the-art performance in various NLP tasks, including text classification, question answering, and natural language inference. RoBERTa further improves BERT by removing the next sentence prediction objective and trains on larger mini-batches (Liu et al., 2019). The massive T5 transformer model (Raffel et al., 2020) studies extensively the variations in training hyperparameters, such as batch size and learning rates, and is trained on a larger corpus than BERT.

With generation objectives Instead of optimizing the prediction of the masked words (Figure 2.3 left), Radford et al. (2018) propose generative pretraining (GPT). It predicts the tokens auto-regressively and optimizes directly with generation loss (Figure 2.3 right). The GPT model can be directly fine-tuned on language generation tasks such as machine translation and question answering. However, GPT only conditions on the leftward contexts and does not learn bidirectional interactions. Lewis et al. (2020) present BART, a denoising auto-encoder for pretraining sequence-to-sequence models. BART is trained by learning to reconstruct the original text from corrupting text with arbitrary noising functions. It uses a standard Transformer-based encoder-decoder architecture with the same bidirectional encoder as BERT.

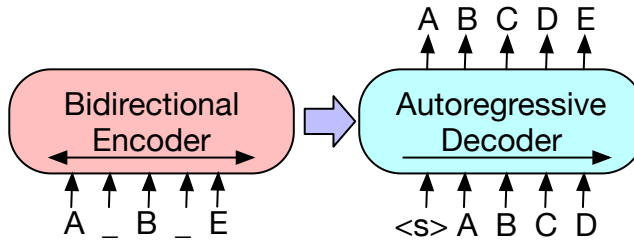


Figure 2.4: The inputs of BART does not need to be aligned with decoder outputs, allowing arbitrary noise transformations. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an auto-regressive decoder.

BART matches the performance of RoBERTa on GLUE and SQuAD, and achieves new state-of-the-art results on a range of language generation tasks, including abstractive dialogue, question answering, and summarization tasks. This demonstrates the benefits of generative pretraining against pretraining with prediction loss.

2.3.4 Cross-lingual Language Models

Pretrained large language models demonstrate their strong ability in capturing semantics and syntactic structures from extremely large unlabelled corpora and benefit downstream NLP tasks through transferring the learned knowledge via fine-tuning. However, languages other than English have much less data resources than English. For low-resource languages, the text data cannot boost the training for large language models.

To address this issue, [Devlin et al. \(2019\)](#) introduce to pretrain a multilingual BERT (mBERT) on the Wikipedia pages with a mixture of 104 languages. All the tokens are encoded with a shared vocabulary. It does not use any marker denoting the input language and does not have any explicit mechanism to encourage translation equivalent pairs to have similar representations. [Lample and Conneau \(2019\)](#) extend the idea of cross-lingual pretraining and pretrain the shared language model with a combination of objectives: 1) the causal language modeling objective which optimizes the probability of a word given the previous words; 2) the masked language modeling objective same as BERT; and 3) the translation language modeling objective which leverages existing parallel bilingual data and learns to align low-resource languages with English. XLM obtains state-of-the-art results on cross-lingual classification, unsupervised and supervised machine translation tasks.

BART demonstrates the benefits of denoising pretraining with generation objectives, [Liu et al. \(2020\)](#) further extends the model by pretraining on a mixture of 25 languages with the same denoising techniques and reconstruction loss. mBART achieves state-of-the-art performance on low-resource machine translations.

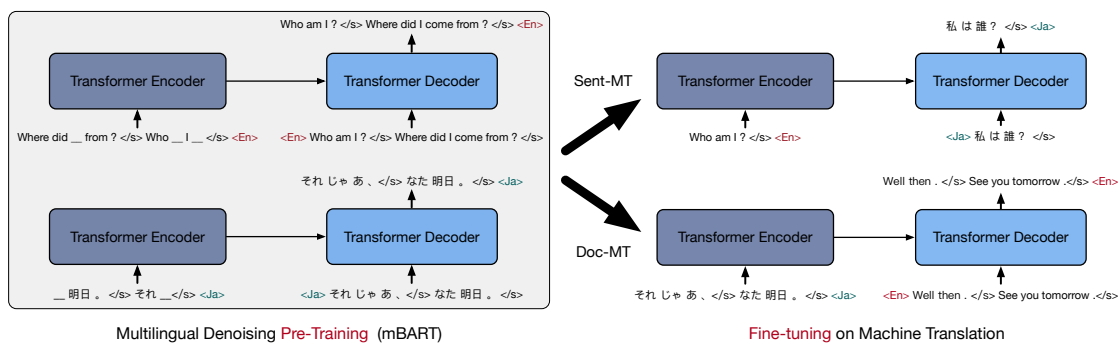


Figure 2.5: The pretraining framework of Multilingual BART (left), where two corruption methods are used (1) sentence permutation (2) word-span masking as the injected noise; and fine-tuning on downstream MT tasks (right). A special language id token is added at both the encoder and decoder.

Chapter 3: Uncover Interpretable Senses from Words

In Section 2.2.2, we present the word2vec models which learn word embeddings that are widely applied in various NLP tasks. In this Chapter, we extend the word2vec model by explicitly revealing word senses and learning sense-specific embeddings. Compared to existing sense-specific embedding models, we focus on ensuring the interpretability of the learned sense embeddings via differentiable hard sense selection. And we expose the discrepancy between computer-centric tasks such as word similarity tasks and human interpretability and propose a new coherence evaluation for sense embeddings that examine how well the learned sense inventory by sense embeddings align with human judgments.

3.1 The necessity of human interpretable sense representations

Word embeddings such as Word2vec (Mikolov et al., 2013a,b) and GloVe (Pennington et al., 2014) capture semantic relations of words from unlabeled raw texts and have been widely applied in various NLP tasks such as information retrieval (Zuccon et al., 2015). However, words also have hidden structures. Many words have multiple senses; for example, the English word "bank" can refer to a financial institution that accepts deposits from customers and makes loans or the side of a river. ELMo (Peters et al., 2018) and pretrained language models like BERT (Devlin et al., 2019) address this issue implicitly by capturing senses in the context of words. Rather than one-size-fits-all word vectors that ignore the nuance of how words are used in context, these new representations have topped the leaderboards for question answering, inference, and classification. Contextual representations have supplanted multi-sense embeddings (Camacho-Collados and Pilehvar, 2018). While these methods learn a vector for *each sense*, they do not encode meanings in downstream tasks as well as contextual representations (Peters et al., 2018).

However, computers are not the only consumer of text representations. Humans also use word representations to understand diachronic drift (Hamilton et al., 2016a), investigate a language’s sense inventory, or to cluster and explore documents. Thus, a primary goal for explicitly revealing the sense structures in words and learning sense-specific word embeddings is to facilitate *human* understanding of word meanings. Unfortunately, multi-sense models have only been evaluated on *computer-centric* dimensions and have ignored the question of *sense interpretability*.

We first develop measures for how well models encode and explain a word’s meaning to a human (3.3). Existing multi-sense models do not necessarily fare best on this evaluation; our proposed model (Gumbel Attention for Sense Induc-

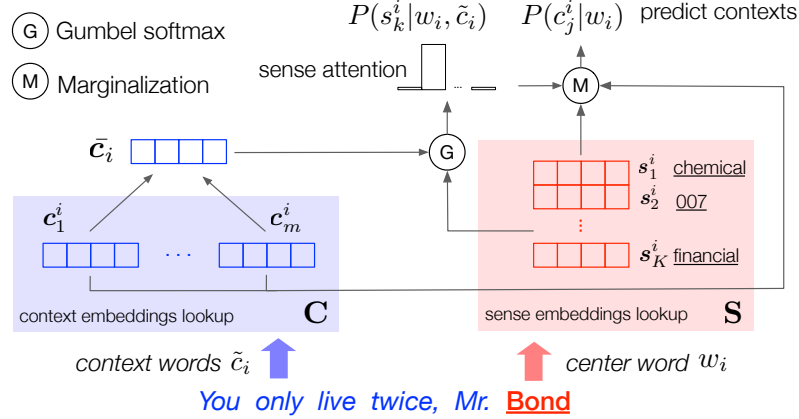


Figure 3.1: Network structure with an example of our GASI model which learns a set of global context embeddings \mathbf{C} and a set of sense embeddings \mathbf{S} .

tion: GASI, 3.2) that focuses on discrete sense selection can better capture human-interpretable representations of senses; comparing against traditional evaluations (3.4), GASI has better contextual word similarity and competitive non-contextual word similarity. Finally, we discuss the connections between representation learning and how modern contextual representations could better capture interpretable senses (3.5). We show that explicitly revealing the inner structure of data-driven models can generate human interpretable representations.

3.2 Attentional Sense Induction

Before we explore human interpretability of sense induction, we first describe our models to disentangle senses from words and learn sense-specific embeddings. Our two models are built on Word2Vec (Mikolov et al., 2013a,b), which we review in the background section 2.2.2. Both models use a straightforward attention mechanism to select which sense is used in a token’s context, which we contrast to alternatives for sense selection (3.2.2.1). Building on these foundations, we introduce our model, GASI, and along the way introduce a soft-attention stepping-stone (SASI).

3.2.1 Foundation: Gumbel Softmax

As we introduce word senses, our model will need to select *which* sense is relevant for a context. The Gumbel softmax (Jang et al., 2016; Maddison et al., 2016) approximates the sampling of discrete random variables; we use it to select the sense. Given a discrete random variable X with $P(X = k) \propto \alpha_k$, $\alpha_k \in (0, \infty)$, the Gumbel-max (Gumbel and Lieblein, 1954) refactors the sampling of X into

$$X = \arg \max_k (\log \alpha_k + g_k), \quad (3.1)$$

where the Gumbel noise $g_k = -\log(-\log(u_k))$ and u_k are i.i.d. from $\text{Uniform}(0, 1)$. The Gumbel softmax approximates sampling $\text{one_hot}(\arg \max_k (\log \alpha_k + g_k))$ by

$$y_k = \text{softmax}((\log \alpha_k + g_k)/\tau). \quad (3.2)$$

Unlike soft selection of senses, the Gumbel softmax can make harder selections, which will be more interpretable to humans.

3.2.2 Reveal Word Senses in Skip-Gram Model

The Word2Vec Skip-Gram model jointly learns word embeddings $\mathbf{W} \in \mathbb{R}^{|V| \times d}$ and context embeddings $\mathbf{C} \in \mathbb{R}^{|V| \times d}$. More specifically, given a vocabulary V and embedding dimension d , it maximizes the likelihood of the context words c_j^i that surround a given center word w_i in a context window \tilde{c}_i ,

$$J(\mathbf{W}, \mathbf{C}) \propto \sum_{w_i \in V} \sum_{c_j^i \in \tilde{c}_i} \log P(c_j^i | w_i; \mathbf{W}, \mathbf{C}). \quad (3.3)$$

As we explained in 3.1, revealing word sense and learning sense-specific embeddings can help with human-centric tasks such as building sense inventories. In the following sections, we discuss existing alternatives to induce sense from contexts and explain our choice—differentiable hard attention via scaled Gumbel Softmax.

3.2.2.1 Why Attention? Musing on Alternatives

For fine-grained sense inventories, it makes sense to have graded assignment of tokens to senses (Erk et al., 2009; Jurgens and Klapaftis, 2015). However, for coarse senses—except for humor (Miller et al., 2017)—words typically are associated with a *single sense*, often a single sense per discourse (Gale et al., 1992). A good model should respect this. Previous models either use non-differentiable objectives or—in the case of the current state of the art, MUSE (Lee and Chen, 2017)—reinforcement learning to select word senses. By using Gumbel softmax, our model both approximates discrete sense selection and is differentiable.

As we argue in the next section, applications with a human in the loop are best assisted by discrete senses; the Gumbel softmax, which we develop for our task here, helps us discover these discrete senses.

3.2.2.2 Attentional Sense Induction

Embeddings We learn a context embedding matrix $\mathbf{C} \in \mathbb{R}^{|V| \times d}$ and a sense embedding tensor $\mathbf{S} \in \mathbb{R}^{|V| \times K \times d}$. Unlike previous work (Neelakantan et al., 2014; Lee and Chen, 2017), no extra embeddings are kept for sense induction.

Number of Senses For simplicity and consistency with previous work, our model has K fixed senses. Ideally, if we set a large number of K , with a perfect pruning

strategy, we can estimate the number of senses per type by removing duplicated senses.

However, this is challenging (McCarthy et al., 2016); instead we use a simple pruning strategy. We estimate a pruning threshold λ by averaging the estimated duplicate sense and true neighbor distances,

$$\lambda = \frac{1}{2}(\text{mean}(D_{dup}) + \text{mean}(D_{nn})), \quad (3.4)$$

where D_{dup} are the cosine distances for duplicated sense pairs and D_{nn} is that of true neighbors (different types). We sample 100 words and if two senses are top-5 nearest neighbors of each other, we consider them duplicates.

After pruning duplicated senses with λ , we can retrain a new model with estimated number of senses for each type by masking the sense attentions. Results in Table 3.2 and 3.5 validate our pruning strategy.

Sense Attention in Objective Function Assuming a center word w_i has senses $\{s_1^i, s_2^i, \dots, s_K^i\}$, the original Skip-Gram likelihood becomes a marginal distribution over all senses of w_i with sense induction probability $P(s_k^i | w_i)$; we focus on the disambiguation given local context \tilde{c}_i and estimate $P(s_k^i | w_i) \approx P(s_k^i | w_i, \tilde{c}_i)$; and thus,

$$P(c_j^i | w_i) \approx \sum_{k=1}^K P(c_j^i | s_k^i) \underbrace{P(s_k^i | w_i, \tilde{c}_i)}_{\text{attention}}, \quad (3.5)$$

Replacing $P(c_j^i | w_i)$ in Equation 3.3 with Equation 3.5 gives our objective function $J(\mathbf{S}, \mathbf{C}) \propto$

$$\sum_{w_i \in V} \sum_{c_j^i \in \tilde{c}_i} \log \sum_{k=1}^K P(c_j^i | s_k^i) P(s_k^i | w_i, \tilde{c}_i). \quad (3.6)$$

Modeling Sense Attention We can model the *contextual sense induction distribution* with soft attention. We call the resulting model soft-attention sense induction (SASI). Although it is a stepping stone to our final model, we compare against it in our experiments as it isolates the contributions of hard attention. In SASI, the sense attention is conditioned on the entire local context \tilde{c}_i with softmax:

$$P(s_k^i | w_i, \tilde{c}_i) = \frac{\exp(\bar{\mathbf{c}}_i^\top \mathbf{s}_k^i)}{\sum_{k=1}^K \exp(\bar{\mathbf{c}}_i^\top \mathbf{s}_k^i)}, \quad (3.7)$$

where $\bar{\mathbf{c}}_i$ is the mean of the context vectors in \tilde{c}_i .

3.2.2.3 Scaled Gumbel Softmax for Sense Disambiguation

To learn *distinguishable sense representations*, we implement *hard* attention in our full model, Gumbel Attention for Sense Induction (GASI). While hard attention

is conceptually attractive, it can increase computational difficulty: discrete choices are not differentiable and thus incompatible with modern deep learning frameworks. To preserve differentiability (and to avoid equally complex reinforcement learning), we apply the Gumbel softmax reparameterization trick to our sense attention function (Equation 3.7).

Vanilla Gumbel The discrete sense sampling from Equation 3.7 can be refactored

$$\mathbf{z}^i = \text{one_hot}(\arg \max_k (\bar{\mathbf{c}}_i^\top \mathbf{s}_k^i + g_k)), \quad (3.8)$$

and the hard attention approximated

$$y_k^i = \text{softmax}((\bar{\mathbf{c}}_i^\top \mathbf{s}_k^i + g_k)/\tau). \quad (3.9)$$

Scaled Gumbel Gumbel softmax learns a flat distribution over senses even with low temperatures: the dot product $\bar{\mathbf{c}}_i^\top \mathbf{s}_k^i$ is too small¹ compared to the Gumbel noise g_k . Thus we use a scaling factor β to encourage sparser distributions,²

$$\gamma_k^i = \text{softmax}((\bar{\mathbf{c}}_i^\top \mathbf{s}_k^i + \beta g_k)/\tau), \quad (3.10)$$

and tune it as a hyperparameter. We append GASI- β to the name of models with a scaling factor. This is critical for learning *distinguishable senses* (Figure 3.2, Table 3.2, and Table 3.5). Our **final objective function** for GASI- β is

$$J(\mathbf{S}, \mathbf{C}) \propto \sum_{w_i \in V} \sum_{w_c \in c_i} \sum_{k=1}^K \gamma_k^i \log P(w_c | s_k^i). \quad (3.11)$$

3.3 Evaluating Interpretability

We turn to traditional evaluations of sense embeddings later (Section 3.4), but our focus is on human interpretability. If you show a human the senses, can they understand why a model would assign a sense to that context? This section evaluates whether the representations make sense to human consumers of multisense models.

In the age of BERT and ELMo, these are the dimensions that are most critical for multisense representations. While contextual word vectors are most useful for *computer* understanding of meaning, *humans* often want an overview of word meanings for other tasks.

Sense representations are useful for human-in-the-loop applications. They help understand semantic drift (Hamilton et al., 2016b): how do the meanings of “gay” reflect social progress? They help people learn languages (Noraset et al., 2017): what does it mean when someone says that I “embarrassed” them? They help linguists

¹This is from float32 precision and saturation of $\log(\sigma(\cdot))$

²Normalizing $\bar{\mathbf{c}}_i^\top \mathbf{s}_k^i$ or directly using $\log P(s_k^i | w_i, \bar{\mathbf{c}}_i)$ results in a similar outcome.

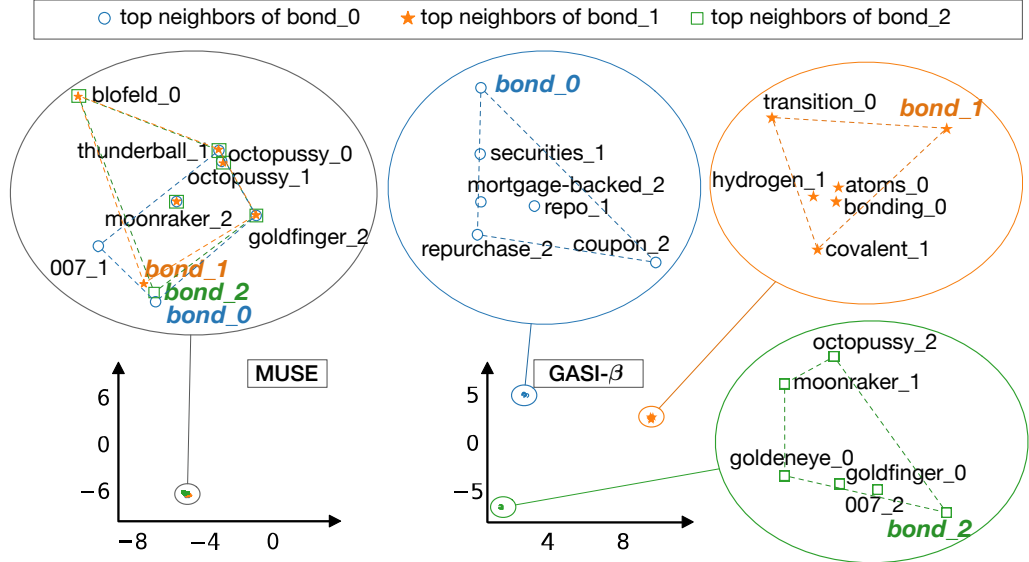


Figure 3.2: t-SNE projections of nearest neighbors for “bond” by *hard-attention* models: MUSE (RL-based) and our GASI- β . Trained on same dataset and vocabulary, both models learn three vectors per word (bond- i is i^{th} sense vector). GASI (right) learns three distinct senses of “bond” while MUSE (left) learns overlapping senses.

Model	Sense Accuracy	Judgment Accuracy	Agreement
MUSE	67.33	62.89	0.73
MSSG-30K	69.33	66.67	0.76
GASI- β	71.33	67.33	0.77

Table 3.1: Word intrusion evaluations on top ten nearest neighbors of sense embeddings. Users find misfit words most easily with GASI- β , suggesting these representations are more interpretable.

understand the sense inventory of a language (Kawahara et al., 2014): what are the frames that can be used by the verb “participate”? These questions (and human understanding) are helped by **discrete senses**, which the Gumbel softmax uncovers.

More broadly, this is the goal of interpretable machine learning (Doshi-Velez and Kim, 2017). While downstream models do not always need an interpretable explanation of *why* a model uses a particular representation, interactive machine learning and explainable machine learning do. To date, multisense representations ignore this use case.

Qualitative analysis Previous papers use nearest neighbors of a few examples to qualitatively argue that their models have captured meaningful senses of words. We also give an example in Figure 3.2, which provides an intuitive view on how the learned senses are clustered by visualizing the nearest neighbors of word “bond”

Model	Accuracy	P	Agreement
MUSE	28.0	0.33	0.68
MSSG-30K	44.5	0.37	0.73
GASI (no β)	33.8	0.33	0.68
GASI- β	50.0	0.48	0.75
GASI- β -pruned	75.2	0.67	0.96

Table 3.2: Human-model consistency on *contextual word sense selection*; P is the average probability assigned by the model to the human choices. GASI- β is most consistent with crowdworkers. Reducing sense duplication by retraining our model with pruning mask improves human-model agreement.

using t-SNE projection (Maaten and Hinton, 2008). Our model (right) disentangles the three sense of “bond” clearly.

However, examples can be cherry-picked. This problem bedeviled topic modeling until rigorous human evaluation was introduced (Chang et al., 2009). We adapt both aspects of their evaluations: *word intrusion* (Schnabel et al., 2015) to evaluate whether individual senses are coherent and *topic intrusion*—rather sense intrusion in this setting—to evaluate whether humans agree with models’ sense assignments *in context*. Using crowdsourced evaluations from Figure-Eight, we compare our models with two previous state-of-the-art sense embeddings models, i.e., MSSG (Neelakantan et al., 2014) and MUSE (Lee and Chen, 2017).³

3.3.1 Word Intrusion for Sense Coherence

Schnabel et al. (2015) suggest a “good” word embedding should have coherent neighbors and evaluate coherence by *word intrusion*. They present crowdworkers four words: three are close in embedding space but one is an “intruder”. If the embedding makes sense, contributors will easily spot the word that “does not belong”.

Similarly, we examine the coherence of ten nearest neighbors of senses in the *contextual word sense selection* task (Section 3.3.2) and replace one neighbor with an “intruder”. We generate three intruders for each sense and collect three judgments per intruder. To account for variation in users and intruders, we count an instance as “correct” if two or more crowdworkers correctly spot the intruder.

Like Chang et al. (2009), we want the “intruder” to be about as frequent as the target but not too similar. For sense s_i^m of word type w_i , we randomly select a word from the neighbors of *another* sense s_i^n of w_i .

All models have comparable model accuracy. GASI- β learns senses that have the highest coherence while MUSE learns mixtures of senses (Table 3.1).

We use the aggregated confidence score provided by Figure-Eight to estimate the level of **inter-rater agreement** between multiple contributors (Figure Eight,

³MSSG has two settings; we run human evaluation with MSSG-30K which has higher correlation with MaxSimC on SCWS.

		MUSE	MSSG	GASI- β
word	agree	4.78	0.39	1.52
overlap	disagree	5.43	0.98	6.36
Glove	agree	0.86	0.33	0.36
cosine	disagree	0.88	0.57	0.81

Table 3.3: Similarities of human and model choices when they agree and disagree for two metrics: simple word overlap (top) and Glove cosine similarity (bottom). Humans agree with the model when the senses are distinct.

The real <u>question</u> is - how are those four years used and what is their value as training?	
s1	hypothetical, unanswered, topic, answered, discussion, yes/no, answer, facts
s2	toss-up, answers, guess, why, answer, trivia, caller, wondering, answering
s3	argument, contentious, unresolved, concerning, matter, regarding, debated, legality

Table 3.4: A case where MSSG has low overlap but confuses raters (agreement 0.33); the model chooses s1.

2018). The agreement is high for all models, and GASI- β has the highest agreement, suggesting that the senses learned by GASI- β are easier to interpret.

3.3.2 Contextual Word Sense Selection

The previous task measures whether individual senses are coherent. Now we evaluate models’ disambiguation of senses *in context*.

Task Description Given a target word in context, we ask a crowdworker to select which sense group best fits the sentence. Each sense group is described by its top ten distinct nearest neighbors, and the sense group order is shuffled.

Data Collection We select fifty nouns with five sentences from SemCor 3.0 (Miller et al., 1994). We first filter all word types with fewer than ten sentences and select the fifty most polysemous nouns from WordNet (Miller and Fellbaum, 1998) among the remaining senses. For each noun, we randomly select five sentences.

Metrics For each model, we collect three judgments for each question. We consider a model correct if at least two crowdworkers select the same sense as the model.

Sense Disambiguation and Interpretability If humans consistently pick the same sense as the model, they must first understand the choices, thus implying the nearest neighbor words were coherent. Moreover, they also agree that among those senses, that sense was the right choice for this token. GASI- β selections are most consistent with humans’; its selections have the highest accuracy and assigns the largest probability assigned to the human choices (Table 3.2). Thus, GASI- β produces sense embeddings that are both more interpretable and distinguishable. GASI without a scaling factor, however, has low consistency and flat sense distribution.

Model Confidence However, some contexts are more ambiguous than others. For fine-grained senses, best practice is to use graded sense assignments (Erk et al., 2013). Thus, we also show the model’s probability of the top human choice; distributions close to $\frac{1}{K}$ (0.33) suggest the model learns a distribution that cannot disambiguate senses. We consider granularity of senses further in sec:related.

Inter-rater Agreement We use the confidence score computed by Figure-Eight to estimate the raters’ agreement for this task. GASI- β has the highest human-model agreement, while both MUSE and GASI without scaling have the lowest.

Error Analysis Next, we explore why crowdworkers disagree with the model even though the senses are interpretable (Table 3.1). Is it that the model has learned *duplicate* senses that both the users and model cannot distinguish (the senses are all bad or identical) or is it that crowdworkers agree with each other but *disagree* with the model (the model selects bad senses)?

Two trends suggest duplicate senses cause disagreement both for humans with models and humans with each other. For two measures of sense similarity—simple word overlap and GLoVe similarity—similarity is lower when users and models agree (Table 3.3). Humans also agree with each other more. For GASI- β , pairs with perfect agreement have a word overlap of around 2.5, while the senses with lowest agreement have overlap around 5.5.

To reduce duplicated senses, we retrain the model with pruning (Section 3.2.2.2, Equation 3.4). We remove a little more than one sense per type on average. To maintain the original setting, for word types that have fewer than three senses left, we compute the nearest neighbors to dummy senses represented by random embeddings. Our model trained with pruning mask (GASI- β -pruned) reaches very high inter-rater agreement and higher human-model agreement than models with a fixed number of senses (Table 3.2, bottom).

3.4 Word Similarity Evaluation

GASI and GASI- β are interpretable, but how do they fare on standard word similarity tasks?

Contextual Word Similarity Tailored for sense embedding evaluation, Stanford Contextual Word Similarities (Huang et al., 2012, SCWS) has 2003 word pairs tied to context sentences. These tasks assign a pair of word types (e.g., “green” and “buck”) a similarity/relatedness score. Moreover, both words in the pair have an associated context. These contexts disambiguate homonymous and polysemous word types and thus captures sense-specific similarity. Thus, we use this dataset to tune our hyperparameters, comparing Spearman’s rank correlation ρ between embedding similarity and the gold similarity judgments: higher scores imply the model captures semantic similarities consistent with the trusted similarity scores.

To compute the word similarity with senses we use two metrics (Reisinger and Mooney, 2010) that take context and sense disambiguation into account: **MaxSimC** computes the cosine similarity $\cos(s_1^*, s_2^*)$ between the two most probable senses s_1^* and s_2^* that maximizes $P(s_k^i | w_i, \tilde{c}_i)$. **AvgSimC** weights average similarity over the combinations of all senses $\sum_{i=1}^K \sum_{j=1}^K P(s_i^1 | w_1, \tilde{c}_1) P(s_j^2 | w_2, \tilde{c}_2) \cos(s_i^1 s_j^2)$.

We compare variants of our model with existing sense embedding models (Table 3.5), including two previous SOTAs: the clustering-based Multi-Sense Skip-Gram model (Neelakantan et al., 2014, MSSG) on AvgSimC and the RL-based Modularizing Unsupervised Sense Embeddings (Lee and Chen, 2017, MUSE) on MaxSimC. GASI better captures similarity than SASI, corroborating that hard attention aids word sense selection. GASI without scaling has the best MaxSimC; however, it learns a flat sense distribution (Figure 3.2). GASI- β has the best AvgSimC and a competitive MaxSimC. While MUSE has a higher MaxSimC than GASI- β , it fails to distinguish senses as well (Figure 3.2, Section 3.3).

We also evaluate the retrained model with pruning mask on this dataset. GASI- β -pruned has the same AvgSimC as GASI- β and higher local similarity correlation (Table 3.5, bottom), validating our pruning strategy (Section 3.2.2.2).

Word Sense Selection in Context SCWS evaluates models’ sense selection indirectly. We further compare GASI- β with previous SOTA, MSSG-30K and MUSE, on the Word in Context dataset (Pilehvar and Camacho-Collados, 2018, WiC) which requires the model to identify whether a word has the same sense in two contexts. To reduce the variance in training and to focus on evaluating the sense selection module, we use an evaluation suited for unsupervised models: if the model selects different sense vectors given contexts, we mark that the word has different senses.⁴ For MUSE, MSSG and GASI- β , we use each model’s sense selection module; for DeConf (Pilehvar and Collier, 2016) and sw2v (Mancini et al., 2017), we follow Pilehvar and Camacho-Collados (2018) and Pelevina et al. (2016) by selecting the closest sense vectors to the context vector. DeConf results are comparable to supervised results (59.4 ± 0.7). GASI- β has the best result (55.3) apart from DeConf itself (58.55), which uses the same sense inventory (Miller and Fellbaum, 1998, WordNet) as WiC.

Non-Contextual Word Similarity While contextual word similarity is best suited for our model and goals, other datasets without contexts (i.e., only word

⁴For monosemous or out of vocab words, we choose randomly.

Model	MaxSimC	AvgSimC
Huang et al. (2012)-50d	26.1	65.7
MSSG-6K	57.3	69.3
MSSG-30K	59.3	69.2
Tian et al. (2014)	63.6	65.4
Li and Jurafsky (2015)	66.6	66.8
Qiu et al. (2016)	64.9	66.1
Bartunov et al. (2016)	53.8	61.2
MUSE_Boltzmann	67.9	68.7
SASI	55.1	67.8
GASI (w/o scaling)	68.2	68.3
GASI- β	66.4	69.5
GASI- β -pruned	67.0	69.5

Table 3.5: Spearman’s correlation 100ρ on SCWS (trained on 1B token, 300d vectors except for Huang et al.). GASI and GASI- β both can disambiguate the sense and correlate with human ratings. Retraining the model with pruned senses further improves local similarity correlation.

pairs and a rating) are both larger and ubiquitous for word vector evaluations. To evaluate the semantics captured by each sense-specific embeddings, we compare the models on non-contextual word similarity datasets.⁵ Like Lee and Chen (2017) and Athiwaratkun et al. (2018), we compute the word similarity based on senses by **MaxSim** (Reisinger and Mooney, 2010), which maximizes the cosine similarity over the combination of all sense pairs and does not require local contexts,

$$\text{MaxSim}(w_1, w_2) = \max_{0 \leq i \leq K, 0 \leq j \leq K} \cos(s_i^1, s_j^2). \quad (3.12)$$

GASI- β has better correlation on three datasets, is competitive on the rest (Table 3.6), and remains competitive without scaling. GASI is better than MUSE, the other hard-attention multi-prototype model, on six datasets and worse on three. Our model can reproduce word similarities as well or better than existing models through our sense selection.⁶

3.4.1 Word Similarity vs. Interpretability

Word similarity tasks (Section 3.4) and human evaluations (Section 3.3) are inconsistent. GASI, GASI- β and MUSE are all competitive in word similarity (Table 3.5

⁵RG-65 (Rubenstein and Goodenough, 1965); SimLex-999 (Hill et al., 2015); WS-353 (Finkelstein et al., 2002); MEN-3k (Bruni et al., 2014); MC-30 (Miller and Charles, 1991); YP-130 (Yang and Powers, 2006); MTurk-287 (Radinsky et al., 2011); MTurk-771 (Halawi et al., 2012); RW-2k (Luong et al., 2013)

⁶Given how good PDF-GM is, it could do better on contextual word similarity even though it ignores senses. Average and MaxSim are equivalent for this model; it ties GASI- β .

Dataset	MUSE	SASI	GASI	GASI- β	PFT-GM
SimLex-999	39.61	31.56	40.14	41.68	40.19
WS-353	68.41	58.31	68.49	69.36	68.6
MEN-3k	74.06	65.07	73.13	72.32	77.40
MC-30	81.80	70.81	82.47	85.27	74.63
RG-65	81.11	74.38	77.19	79.77	79.75
YP-130	43.56	48.28	49.82	56.34	59.39
MT-287	67.22	64.54	67.37	66.13	69.66
MT-771	64.00	55.00	66.65	66.70	68.91
RW-2k	48.46	45.03	47.22	47.69	45.69

Table 3.6: Spearman’s correlation on non-contextual word similarity (MaxSim). GASI- β has higher correlation on three datasets and is competitive on the others. PFT-GM is trained with two components/senses while other models learn three. A full version including MSSG is in appendix.

and Table 3.6), but only GASI- β also does well in the human evaluations (Table 3.2). Both GASI without scaling and MUSE fail to learn distinguishable senses and cannot disambiguate senses. High word similarities do not necessarily indicate “good” sense embeddings quality; our human evaluation—*contextual word sense selection*—is complementary.

3.5 Related Work: Representation, Evaluation

Schütze (1998) introduces context-group discrimination for senses and uses the centroid of context vectors as a sense representation. Other work induces senses by context clustering (Purandare and Pedersen, 2004) or probabilistic mixture models (Brody and Lapata, 2009). Reisinger and Mooney (2010) first introduce multiple sense-specific vectors for each word, inspiring other multi-prototype sense embedding models. Generally, to address polysemy in word embeddings, previous work trains on annotated sense corpora (Iacobacci et al., 2015; Gómez-Pérez and Denaux, 2019) or external sense inventories (Labutov and Lipson, 2013; Chen et al., 2014; Jauhar et al., 2015; Chen et al., 2015; Wu and Giles, 2015; Pilehvar and Collier, 2016; Mancini et al., 2017); Rothe and Schütze (2017) extend word embeddings to lexical resources without training; others induce senses via multilingual parallel corpora (Guo et al., 2014; Šuster et al., 2016; Ettinger et al., 2016).

We contrast our GASI to *unsupervised* monolingual multi-prototype models along two dimensions: *sense induction methodology* and *differentiability*. Our focus is unsupervised induction because for interpretability to be useful, we assume that sense inventories and disambiguations are either unavailable or imperfect.

On the dimension of *sense induction methodology*, Huang et al. (2012) and Neelakantan et al. (2014) induce senses by context clustering; Tian et al. (2014) model a corpus-level sense distribution; Li and Jurafsky (2015) model the sense assignment as a Chinese Restaurant Process; Qiu et al. (2016) induce senses by

minimizing an energy function on a context-depend network; [Bartunov et al. \(2016\)](#) model the sense assignment as a steak-breaking process; [Nguyen et al. \(2017\)](#) model the sense embeddings as a weighted combination of topic vectors with pre-computed weights by topic models; [Athiwaratkun et al. \(2018\)](#) model word representations as Gaussian Mixture embeddings where each Gaussian component captures different senses; [Lee and Chen \(2017\)](#) compute sense distribution by a separate set of sense induction vectors. The proposed GASI marginalizes the likelihood of contexts over senses and induces senses by local context vectors; the most similar sense selection module is a bilingual model ([Šuster et al., 2016](#)) except that it does not introduce lower bound for negative sampling but uses weighted embeddings, which results in mixed senses.

On the dimension of *differentiability*, most sense selection models are *non-differentiable* and discretely select senses, with two exceptions: [Šuster et al. \(2016\)](#) use weighted vectors over senses; [Lee and Chen \(2017\)](#) implement hard attention with RL to mitigate the non-differentiability. In contrast, GASI keeps full differentiability by reparameterization and approximates discrete sense sampling with the scaled Gumbel softmax.

However, the elephants in the room are BERT and ELMo. While there are specific applications where humans might be better served by multisense embeddings, computers seem to be consistently better served by contextual representations. A natural extension is to use the aggregate representations of word senses from these models. Particularly for ELMo, one could cluster individual mentions ([Chang, 2019](#)), but this is unsatisfying at first blush: it creates clusters more specific than senses. BERT is even more difficult: the transformer is a dense, rich representation, but only a small subset describes the meaning of individual words. Probing techniques ([Perone et al., 2018](#)) could help focus on semantic aspects that help *humans* understand word usage.

3.5.1 Granularity

Despite the confluence of goals, there has been a disappointing lack of cross-fertilization between the traditional knowledge-based lexical semantics community and the representation-learning community. Following the trends of sense learning models, we—from the perspective of those used to VerbNet or WordNet—use far too few senses per word. While there is disagreement about sense inventory, “hard” and “line” ([Leacock et al., 1998](#)) definitely have more than three senses. Expanding to granular senses presents both challenges and opportunities for future work.

While moving to a richer sense inventory is valuable future work, it makes human annotation more difficult ([Erk et al., 2013](#))—while we can expect humans to agree on which of three senses are used, we cannot for larger sense inventories. In topic models, [Chang et al. \(2009\)](#) develop topic log odds (in addition to the more widely used model precision) to account for graded assignment to topics. Richer user models would need to capture these more difficult decisions.

However, moving to more granular senses requires richer modeling. Bayesian nonparametrics ([Orbanz and Teh, 2010](#)) can determine the number of clusters that

best explain the data. Combining online stick breaking distributions (Wang et al., 2011) with GASI’s objective function could remove unneeded complexity for word types with few senses and consider the richer sense inventory for other words.

3.6 Conclusion

The goal of multi-sense word embeddings is not just to win word sense evaluation leaderboards. Rather, they should also *describe* language: given millions of tokens of a language, what are the patterns in the language that can help a lexicographer or linguist in day-to-day tasks like building dictionaries or understanding semantic drift. Our differentiable Gumbel Attention Sense Induction (GASI) offers comparable word similarities with multisense representations while also learning more distinguishable, interpretable senses.

However, simply asking whether word senses look good is only a first step. A sense induction model designed for human use should be closely integrated into that task. While we use a Word2Vec-based objective function in Section 3.2, ideally we should use a human-driven, task-specific metric (Feng and Boyd-Graber, 2019) to guide the selection of senses that are distinguishable, interpretable, **and useful**.

Chapter 4: Human-knowledge Guided Automatic Song Translation for Tonal Languages

Large pretrained language models have brought huge gains in generation tasks in low-resource domains (Section 2.3.3). However, in real applications, aligning the generated content with user expectations is crucial. For example, to compose compelling poems, the content must adhere to user-expected formats and themes. In other cases, such as translating song lyrics, the output must also follow specific musical rules, while the parallel data for fine-tuning is even scarcer than that for poetry. In this Chapter, we incorporate rules from human knowledge as inductive bias in one specific application—automatic song translation (AST) for tonal languages—and provide interpretable controls on the generated contents.

In fact, the AST for tonal languages has the unique challenge of aligning words’ tones with the melody of a song *in addition to* conveying the original meaning. We propose three criteria for effective AST—preserving semantics, singability and intelligibility—and develop objectives for these criteria. And we develop a new benchmark for English–Mandarin song translation and develop an unsupervised AST system, the Guided AliGnment for Automatic Song Translation (GagaST), which combines pre-training with three decoding constraints. Both automatic and human evaluations show GagaST successfully balances semantics and singability.¹ This demonstrates the value of incorporating human knowledge into the data-driven generation model.

4.1 Introduction

Suppose you are asked to translate the lyrics “let it go” from the Disney musical *Frozen* into Mandarin Chinese. Some good, literal translations of this would be A) “fàng shǒu”, B) “fàng shǒu ba” or C) “ràng tā qù ba” (Figure 4.1); these get the meaning across and are the domain of traditional machine translation. However, what if you needed to sing this song in Chinese? These literal translations simply do not work: translation A) and C) do not match the number of notes and break the original rhythm; while the tone of translation B) does not match with the pitch flow of the original melody.

Song translation, unlike lyrics translation (subtitling), aims to translate the lyrics so that it can be sung with the original melody. Therefore, the translated

¹We illustrate the task and examples of translated songs by GagaST on <https://gagast.github.io/posts/gagast>.

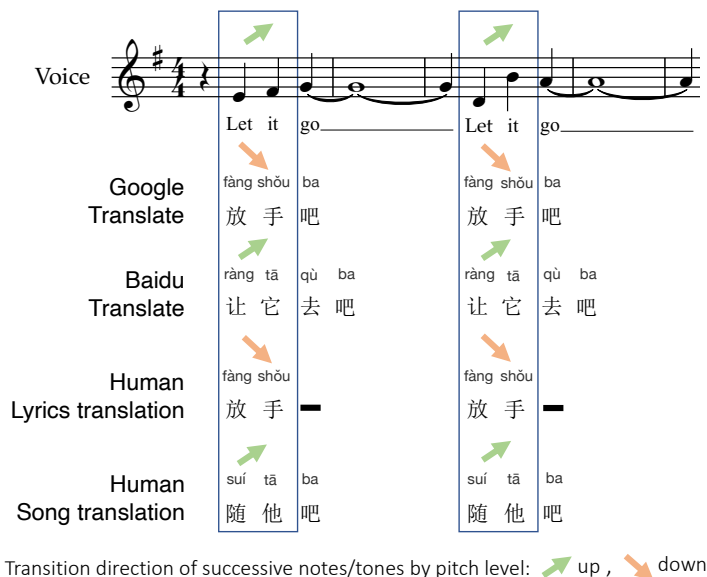


Figure 4.1: Example Mandarin translations for “Let it go” in *Frozen*. Of these, only the official human song translation considers whether a singer could sing the song: it fits the length of the notes and matches the tones with the pitch of notes.

lyrics must match the prosody of the preexisting music in addition to retaining the original meaning. In *Singable Translations of Songs*, Low (2003) says, this is an uncommon and an unusually complex task, a translator must bear in mind the rhythms, note-values, phrasings, and stresses. Nonetheless, there are cultural and commercial incentives for more efficient song translation; *Frozen* alone made over a half a billion dollars in non-English box office receipts² and *Les Misérables* (musical) has been performed in over a dozen languages on stage.

As we discuss in Section 4.2, while translating Western songs resembles poetry translation, translating into *tonal* languages (e.g., Mandarin, Zulu and Vietnamese) brings new problems. In tonal languages, a word’s pitch contributes to its meaning (Figure 4.2); when singing in tonal languages, the tones of translated words must align with the “flow” of the pitches in the music (Section 4.2.1). For example, if “fáng shǒu” were sung instead of “fàng shǒu” (because notes are going up), a listener might hear “defensive” instead of the intended meaning.

This paper builds the first system for automatic song translation (AST) for one tonal language—Mandarin. Section 4.3 proposes three criteria—*preserving semantics*, *singability* and *intelligibility*—needed in an AST system.

Guided by those goals, we propose an unsupervised AST system, Guided AliGnment for Automatic Song Translation (GagaST). GagaST begins with an out-of-domain translation data (Section 4.4.1) and adds constraints that favor translations that are the appropriate length and whose tones match the underlying music (Section 4.4.3). Naturally, such constraints result in a trade-off between semantic meaning and singability/intelligibility. Section 4.5.4 discusses this trade-off between

²[https://www.the-numbers.com/movie/Frozen-\(2013\)#tab=international](https://www.the-numbers.com/movie/Frozen-(2013)#tab=international)

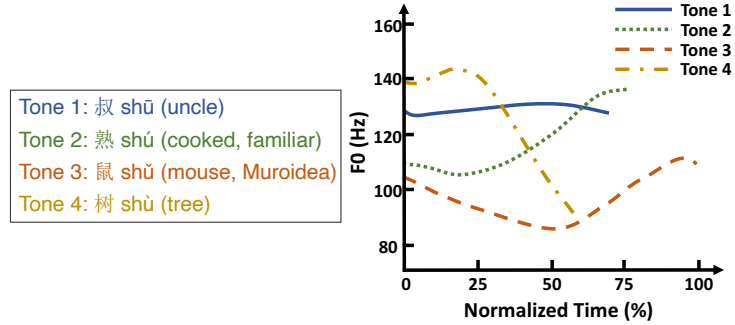


Figure 4.2: In total languages like Mandarin, the pitch changes the meaning of the words (left). Each of the four tones in Mandarin (right) has a different pitch profile. Figure from [Xu \(1997\)](#).

alignment scores and BLEU.

These criteria also form the evaluation for our initial evaluation (Section 4.5.3). However, we go beyond an automatic evaluation through a human-centered evaluation from musicology students. GagaST creates singable songs that make sense given the original text, and our proposed alignment scores correlate with human judgement (Section 4.5.5).

4.2 Background: Prose, Poetry, and Song Translation

The form of written or spoken language has two divisions: prose, which has a natural flow of speech and grammatical structure; and verse, which is typically rhythmic and has special line breaks, such as traditional poetry and song lyrics.

The vast majority of machine translation research has been focused on prose translation and has made huge progress; while verse translation is more difficult as it must obey the rhythmic constraints and is less developed. In his *tour de force* work *Le Ton Beau de Marot*, Douglas Hofstadter created eighty-nine translation of a single poem to capture various aspects of what makes the task difficult ([Hofstadter, 1997](#)).

In western verse, the rhythmic structure are mostly defined by meter, such as the iambic pentameter for sonnets, which defines the length of each line, the patterns of long syllables versus short ones and the stressed ones versus weak ones. Existing work ([Greene et al., 2010](#); [Ghazvininejad et al., 2018](#)) use finite-state constraints to encode both meter and rhyme.

Song translation, on the other hand, can be viewed as a translation where the melody defines the constraints. Reproducing *all* of the essential values of a song—perfectly matching the meaning, perfectly singable, and perfectly understandable—is an impossible ideal ([Franzon, 2008](#)). Thus, tradeoffs are unavoidable. [Low \(2003\)](#) argues for prioritizing *singability* over other qualities such as *sense* and *rhyme* since “effectiveness on stage” is a practical necessity.

Tonal language (e.g., Mandarin, Zulu and Vietnamese) dramatically increases the complexity of singability, and raises a new issue of intelligibility.

	Original lyrics				Misheard lyrics			
Pitch level	66	68	66	65	66	68	66	65
Pronunciation	sì	zài	yǎn	qián	sǐ	zài	yǎn	qián
Lyrics	似	在	眼	前	死	在	眼	前
English translation	as	in	front of	eyes	die	in	front of	eyes
Pitch alignment score	0.5				0.75			

Figure 4.3: A misheard example in Mandarin song caused by a mismatch between music pitch flow and the lyric’s tones. The heard word is “sǐ zài” instead of “sì zài”, because notes are going up and “sì zài” is going down by the sandhi of Mandarin tone.

4.2.1 Song Translation for Tonal Languages

For tonal languages, pitch contributes to the meaning of words. In a conservative estimation, fifty to sixty percent of the world’s languages are tonal (Yip, 2002) and cover over 1.5 billion people. For the lyrics to be *intelligible*, the speech tone and music tone should be correlated (Schneider, 1961). If not, the pitch contour could override the intended tone, which could produce different meanings. This is not just a theoretical consideration; Figure 4.3 shows how lyrics can be and have been misunderstood.³

4.2.2 Mandarin Tones and how to Sing them

Schellenberg (2013) summarizes the rules of singing with tone with a focus on Chinese dialects. The tonal system of Mandarin has two components:

- **The pitch level and shape of tones.** Four Mandarin tones are used since the 19th century. We denote tones with a diacritic over the vowel whose shape roughly matches the shape of the tone. The four tones are a high level (tone 1, e.g., shūo), rising (tone 2, yú), falling-rising (tone 3, wǒ) and falling (tone 4, huài). Their pitch level and shape are shown in Figure 4.2, right.
- **The sandhi of tones.** Some combinations of tones have difficult articulatory patterns, so words that might normally have one tone might take another. For example “nǐ” and “hǎo” are typically both third tone, but when they are together it is pronounced as “ní hǎo”, with the first syllable changing to a *second* tone. These changes are called sandhi (Xu, 1997; Hu, 2017).

Mandarin tones interact with singing in two ways (Yinliu et al., 1983; Schellenberg, 2013) to ensure lyrics are intelligible. First, at a local level, the *shape of tones* of individual characters should be consistent with the musical notes they’re matched with; for example, in “Love Island” (Figure 4.4), “shàng” in the blue block

³Additional misheard examples on demo page
https://gagast.github.io/posts/gagast/#misunderstanding_examples

wǒ yǐ wàng jì wǒ céng huó guò diū le gǎn guān bēi shāng cǐ kè wǒ zhàn zài jīng tāo hài
 我已 忘记 我曾 活过 丢了 感官 悲伤 此刻 我 站 在 惊涛骇
 I have forgotten (that) I've lived. (I've) lost (my) sense (my) sorrow. Right now I'm standing above the terrifying

làng dà hǎi zhī shàng wǒ yǎng tóu wàng wàng xiàng gèn gǔ wú shēng de yuè liang hēi àn
 浪 大海之 上 我 仰 头 望 望 向 亘 古 无 声 的 月 亮 黑 暗
 stormy sea (above). I look up look up to the eternal silence moon. Dark





























Cases: One character (syllable) aligns with a single note

has the “falling” shape and the group of notes that it assigned with also goes falling. Second, and a global level, the music *pitch contour* should align with the tones of the corresponding syllables (taking sandhi into account). In practice, we align the transitions between successive syllables and successive notes (Figure 4.5), based on the idea that tones are relative (Schellenberg, 2013).

This section formally defines automatic song translation (AST) for tonal languages and introduce three criteria for what makes for a good song translation. These criteria form the foundation for the quantitative metrics we use in the experiment.

There are three major criteria that singable song translation needs to fulfil:

- 36

$W_i \backslash W_{i-1}$	Tone 1	Tone 2	Tone 3	Tone 4
Tone 1	  	 	 	
Tone 2	 	 	 	
Tone 3	 	 	 	
Tone 4			 	 






 Leap Up
  Step Up
  Level
  Step Down
  Leap Down

Figure 4.5: For the translated songs in Mandarin to be singable, the transition directions of successive music pitch should align with that of the tones of successive characters. The arrows show the acceptable transition directions (summarized in this paper) in music for two successive Mandarin characters (w_{i-1}, w_i) based on the shape of Mandarin tones and the sandhi of tones.

notes	A3	C4	D4	REST	F4	G4	F4	F4
pitch level	57	60	62		65	67	65	65
duration	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	1	3	$\frac{3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
syllables	How	a-	bout		love?			

Table 4.1: A piece of song “Seasons of love” from the musical *Rent*. We convert the notes into a normalized numerical pitch level for actual computation.

- **Intelligibility.** The translated song need to be understood by the listener. This quality has two components. First, could a listener produce *any* transcription of the lyrics. If the lyrics are too fast or garbled because the keywords do not fit well with the music, the lyrics are unintelligible. Beyond this basic test of recognizability, the lyrics must also be accurate: does this transcription match the intended meaning. Both aspects matter for stage performance, since the audience suppose to understand the content instantly to follow the plot. For pop song covering, not understanding all contents could be acceptable for some audience; however, hilarious misheard lyrics will hurt the experiences (Figure 4.3).

4.3.2 Task Definition

We define the AST task as follows: given an aligned pair of melody M and source lyrics X , generate translated text Y in the target language that align with the input melody M . Each syllable in lyrics may align with one or multiple successive notes (Table 4.1 and Figure 4.4), i.e, each syllable aligns with a group of successive notes with length larger or equal to one.

Specifically, $X = [x_1, \dots, x_L]$ are the input lyrics with L syllables; while the melody M consists of three sequences:

1. The pitch values⁴ of notes $P = [\mathbf{p}_1, \dots, \mathbf{p}_L]$, where $\mathbf{p}_i = [p_i^0, \dots]$ are the pitch values of the i^{th} notes group assigned to syllable x_i , with $|\mathbf{p}_i| \geq 1$;
2. The durations $D = [\mathbf{d}_1, \dots, \mathbf{d}_L]$, where $\mathbf{d}_i = [d_i^0, \dots]$ is the real-valued duration⁵ of each note in the i -th group, with $|\mathbf{d}_i| \geq 1$;
3. $R = [r_1, \dots, r_L]$, where r_i is the real-valued duration of the *REST* note before note group \mathbf{p}_i . If no *REST* exists before \mathbf{p}_i , $r_i = 0.0$.

At each position i , we create an aligned syllable-notes pair $(x_i, (\mathbf{p}_i, \mathbf{d}_i, r_i))$, where the tuple (pitches \mathbf{p}_i , duration \mathbf{d}_i , *REST* info r_i) contains the information needed of the notes group that assigned to syllable x_i .

4.3.3 Aligning Lyrics to Music

To make translated songs singable and intelligible, we summarize three types of critical lyric-melody alignments for English-Mandarin AST (c.f., Section 4.2.1 and 4.2.2), which we use in both our objective functions and evaluation metrics.

For pitch and rhythmic alignment, all the constraints are either uni-gram or bi-gram, for which we design three scores that computes the alignment level of each syllable-notes pair at position i : $S_{ns}^i, S_{pc}^i, S_R^i$. We elaborate details in the following subsections.

4.3.3.1 Length Alignment

The number of syllables L_y in translated lyrics Y needs to fit the number of musical phrases in the melody M , so that it can be sung with the provided music. It is unnecessary to keep the grouping of the original notes in M for the translated song.

4.3.3.2 Pitch Alignment

For tonal languages, the pitches are required to match the translated songs. As described in Section 4.2.2, there are two types of pitch alignments: 1) the *tone shape* of each syllable (Figure 4.4 blue box) should align with the shape of the assigned group of notes; 2) the overall *pitch contour* of the notes should align with the tones of lyrics and in practice, we align the successive transitions of two notes and two syllables.

Tone shape alignment. It is at *single-syllable* level. We only consider this alignment for the syllable that assigns to more than one notes. The shape of the tone is predefined in a tonal language (Wee, 2007), i.e., Mandarin tone shape (Xu, 1997)

⁴1.0 means a semitone

⁵1.0 means a quarter note

can be viewed in Figure 4.2. The shape of notes is the pitch contour of corresponding notes group \mathbf{p}_i . For computing tone shape alignment score in Mandarin and each group of \mathbf{p}_i that assigns to syllable x_i , we estimate its shape by interpolation of the second order on \mathbf{p}_i , and classify it into one of the five categories: level, rising, falling, rising-falling, falling-rising. For example, if $p_{max}^i - p_{min}^i > 1.0$ and the estimated curve is convex with axis in the middle of \mathbf{p}_i , we fit it into category “falling-rising”. Then we compare the shape with that of the syllable y_i , and compute the local tone shape match score S_{ns}^i :

$$S_{ns}^i = \begin{cases} 1.0 & \text{if the shape matches,} \\ \epsilon & \text{if not match,} \end{cases} \quad (4.1)$$

where ϵ is the probability to accept error; “level” can match with any tone, “rising” matches with tone2 (yú), “falling” matches with tone4 (huài), “falling-rising” matches with tone3 (wǒ) while “rising-falling” matches none.

Pitch contour alignment. It compares the transitions between tones (t_{i-1}, t_i) of successive syllables (y_{i-1}, y_i) that belong the same word and the pitch contour of corresponding successive notes $(\mathbf{p}_{i-1}, \mathbf{p}_i)$.⁶ Each transition (the movement from one syllable/note to the next) can be categorized as *level*, *step up*, *leap up*, *step down* and *leap down*. For Mandarin, according to Yinliu et al. (1983), we summarize the acceptable notes’ transitions of two successive characters as illustrated in Figure 4.5. Similarly, for each pair of syllables (y_{i-1}, y_i) , we compute the local pitch contour S_{pc}^i as follow:

$$S_{pc}^i = \begin{cases} 1.0 & \text{if the contour matches,} \\ \epsilon & \text{if not match,} \end{cases} \quad (4.2)$$

where ϵ is the probability to accept error.

4.3.3.3 Rhythmic Alignment with Word Segmentation in Mandarin

A *REST* note represents the interval of silence. For Mandarin, a word should not be broken up by a *REST*, and sometimes *REST*s indicate the end of a phrase and correlate with the punctuation ([punc]), see Figure 4.4 for examples. Therefore, when there is a *REST* note before y_i (after y_{i-1}), i.e., $r_i > 0.0$, we reward the [punc] and word segmentation between y_{i-1} and y_i :

$$S_R^i = \begin{cases} 1.0 & \text{if } r_i > 0.0 \text{ and [punc] after } y_{i-1}, \\ 1.0 & \text{if } r_i = 0.0, \\ P_{seg} & \text{if } r_i > 0.0 \text{ and not [punc],} \\ \epsilon & \text{otherwise.} \end{cases} \quad (4.3)$$

P_{seg} is the probability that (y_i, y_{i-1}) are segmented into different words (the higher the probability, the better it is to have a pause between them), and ϵ is a parameter that represents our tolerance of having a rest within a word.

⁶We considers only the first note p_i^0 in group \mathbf{p}_i if has more than one notes in each group

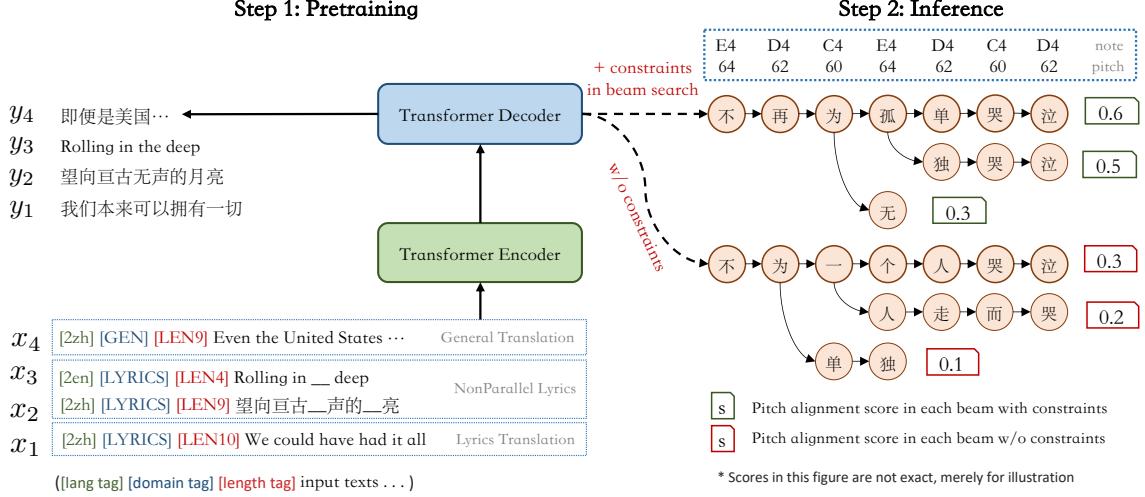


Figure 4.6: Overview of GagaST for English–Mandarin song translation. We first pre-train a lyrics translation model with mixture domain data (left); and then add alignment constraints in decoding scoring function during inference (right), we use unconstrained version as our baseline in the experiment.

4.4 GagaST

To build an AST system for English-Mandarin song translation, ideally, we can learn all alignments by data-driven models with large amount of parallel data, i.e., the aligned triples (M, X, Y) . However, let alone triples, we do not have sufficient accurate parallel data for Mandarin.⁷ In this case, we leverage cross-domain pre-training and propose an unsupervised AST system baseline, Guided AliGnement for Automatic Song Translation (GagaST).

For the pre-training, we collect a large amount of non-parallel lyrics data in both English and Mandarin, as well as a small set of lyrics translation (subtitling) data;⁸ details about training dataset are in Section 4.5.1.

4.4.1 Song-Text Style Translation

To produce faithful translations in song-text style, we pre-train a transformer-based translation model with cross-domain data: translation data in general domain, the collected monolingual lyrics data and a small set of lyrics translation data. We adopt cross-domain training to optimize a translation model that fits into the lyrics domain. We append domain tags (Figure 4.6) before each input entry to control the model to produce translations merely in lyrics domain during song translation. For monolingual lyrics data, we adopt BART pre-training strategy (Lewis et al., 2020).

⁷The only parallel dataset in Mandarin parsed from web contains lots errors in notes and mismatches between syllables and notes; we need accurate alignments for intelligibility

⁸The translations are in plain text, not in lyrics style

4.4.2 Length Control

To meet the length alignments, we pre-define the syllable-notes assignments with two strategies:⁹ 1) *one-to-one*, i.e., for each note, we produce one syllable; 2) *one-to-many*, we use the original notes grouping in the input melody, and assigns one syllable to each note group. In this case, the length of target translation is known. Following Lakew et al. (2019), we use length tag “[LEN\$*i*]” to control the length of outputs during pre-training, where \$*i* refers to the length of the target sequence.

4.4.3 Music Guided Alignment Constraints

There is no available parallel data to learn the lyric-melody alignments with data-driven models, therefore, we impose constraints (Section 4.3.3) in the decoding phase. More specifically, since all constraints that we design are either uni-gram (tone shape, *REST*) or bi-gram (pitch contour, *REST*), we directly apply the lyric-melody alignment constraints at each step of beam search as rewards and penalties in the scoring function :

$$\begin{aligned} \log P(Y | X, M) = & \sum_{i=0}^L [\log P(y_i | y_{i-1:0}, X) \\ & + \lambda_{pc} \log S_{pc}^i + \lambda_{ns} \log S_{ns}^i + \lambda_R \log S_R^i, \end{aligned} \quad (4.4)$$

where S_{pc} , S_{ns} , and S_R refer to the alignment scores for pitch contour, note shape and the rhythm, λ_{pc} , λ_{ns} , and λ_R represent the corresponding hyper-parameters that controls the influence of each constraints.

4.5 Experiments

In this section, we provide the details of data sets, describe the model configuration, introduce the proposed evaluation metrics, analyze the evaluation results, and explain how we handle the trade-offs among different song qualities.

4.5.1 Training Datasets and Model Configuration

WMT translation data We use the news commentary and back-translated news datasets from WMT14, which consisting of about 29.6 million en2zh sentence pairs.

Monolingual lyrics data We collect monolingual lyrics in both Mandarin and English from the web, which contains about 12.4 million lines of lyrics for Mandarin and 109.5 million for English after removing the duplication.

⁹Dynamic mapping between the note sequence and the syllables to be generated increase the search space exponentially.

Lyrics translation data We crawl a small set of lyrics translation data from the web,¹⁰ which contains 140 thousands pairs of English-to-Mandarin lines. These translations are not singable.

We preprocess all data with fastBPE (Sennrich et al., 2016) and a code size of 50,000. We choose standard encoder-decoder Transformer (Vaswani et al., 2017) model with an architecture of 768 hidden units, 12 heads, GELU activation, a dropout rate of 0.1, 512 max input length, 12 layers in encoder, 12 layers in decoder.

4.5.2 Evaluation Dataset

For evaluation, we need aligned triples (melody M , source lyrics X , target reference lyrics Y), where M and X are syllable-to-notes aligned; and the reference Y should be singable and intelligible. Without copyright and accessibility to the singable translated songs, we chose fifty songs from the lyrics translation dataset that have open-source music sheets on the web, and create aligned triples manually. However, the reference lyrics in this dataset do not necessarily resemble song-text style and are not singable, we use them merely to provide a coarse estimation for semantic changes. Twenty songs are used as the validation set (464 lines) and thirty songs as the test set (713 lines).

4.5.3 Evaluation Metrics

An AST system for tonal languages should generate translated songs that are singable and intelligible while conveying the original semantics to the largest extent (Section 4.2 and 4.3.1). Evaluating such system is an intrinsically hard task since all three qualities can be subjective. Especially for semantics, we lack golden references and songs do not have to be as accurate as translation, it is hard to say how many changes in semantics is acceptable. Therefore, we first establish objective evaluations based on expert knowledge and then design human annotation tasks to provide subjective evaluation.

4.5.3.1 Objective Evaluation

Musicians and linguists draw connections between singability and intelligibility with how the melody and lyrics align together (Section 4.2). Based on the expert knowledge, we summarize three metrics that measures lyrics-melody alignments which help to improve singability and intelligibility for Mandarin (Section 4.3.3). We use these metrics as our objective evaluation for singability and intelligibility. Specifically, for **pitch** and **rhythmic alignment**, we normalize the score to 0 – 1.0 by the length of alignment pairs L_i , that is, based on Equation 4.1, 4.2 and 4.3,

$$s_{[\cdot]} = \sum_1^{L_i} S_{[\cdot]}^i / L_i, \quad (4.5)$$

¹⁰<https://lyricstranslate.com/>

Syllable-notes Assignment	Model	Pitch		Rhythm		Length		Semantics BLEU \uparrow
		contour \uparrow	shape \uparrow	avg #	of missed rests \downarrow	longer \downarrow	shorter \downarrow	
one-to-one	GagaST w/o constraints	0.28	-		0.53	9 (0.09)	0	24.0
	GagaST	0.51	-		0.31	26 (0.21)	0	16.9
	-only contour	0.51	-		0.45	26 (0.21)	0	16.8
	-only rest	0.28	-		0.31	11 (0.09)	0	23.8
one-to-many	GagaST w/o constraints	0.29	0.49		0.62	4 (0.12)	0	22.1
	GagaST	0.50	0.55		0.28	13 (0.13)	0	15.9
	-only contour	0.51	0.50		0.42	7 (0.12)	0	15.8
	-only shape	0.29	0.56		0.44	4 (0.12)	0	21.6
	-only rest	0.29	0.49		0.28	5 (0.12)	0	21.6

Table 4.2: Objective results on test set of GagaST with different constraints under one-to-one and one-to-many assignments. All results here use the same pre-training checkpoint and length tags are applied. For length score, 9 (0.09) means that 9 out of 713 samples are longer than the predefined length with an average ratio 0.09.

For **length alignment**, we compute: 1) N_l , the number of samples that has length longer than the predefined length L_i ; 2) N_s , that are shorter than L_i . And for each case, we show the average error ratio of $\{\Delta l_i/L_i\}_1^{N[\cdot]}$.

For **semantics**, despite the fact that we lack golden singable translations, we follow the common practice and calculate the BLEU scores (Papineni et al., 2002) between the translated songs and plan-text translation for reference.

4.5.3.2 Subjective Evaluation

To demonstrate whether the proposed metrics align with actual human experiences and examine the quality of the translated songs by GagaST, we conduct human evaluations. Lacking reliable singing voice synthesizer tools, following Sheng et al. (2021), we show the annotators the music sheets without singing. And we collaborate with the annotation team from Music School to provide expert evaluation.

Specifically, we randomly select five songs from the test set and show the music sheets (see Appendix B) of the first ten sentences of each translated song by GagaST to five annotators.

Following mean opinion score (MOS) (Rec, 1994) in speech synthesizer task, we use five-point scales (1 for bad and 5 for excellent). And we evaluate the songs in four aspects: 1) *sense*, fidelity to the meaning of the source lyric; 2) *style*, whether the translated lyric resembles song-text style; 3) *listenability*, whether the translated lyric sounds melodious with the given melody; 4) *intelligibility*, whether the audience can easily comprehend the translated lyrics if sung with provided melody. The latter two qualities require the annotators to sing the song by themselves.

4.5.4 Hyper-parameters and Trade-offs

The GagaST adds constraints in the decoding scoring functions to enforce lyric-music alignments. There are trade-offs between semantics and other alignments. We analyze the increasing curves of pitch alignment scores against BLEU on valid set,

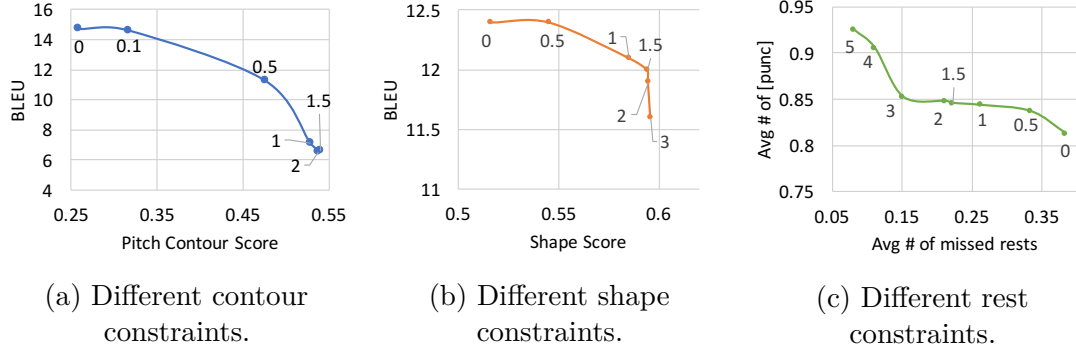


Figure 4.7: Trade-off between semantics and lyric-music alignments; all curves are drawn for the valid set.

and choose the hyper-parameters where the alignment scores increase fast while the BLEU decrease slow. The *REST* constraint does not affect the BLEU (Table 4.2) but the number of punctuation. We should prevent a large increase in the number of punctuation while reducing the mismatches between the *REST* and semantic segmentation. Based on Figure 4.7, we chose: $\lambda_{pc} = 0.5$; $\lambda_{ns} = 1.0$; $\lambda_R = 1.5$.

4.5.5 Evaluation Results

4.5.5.1 Objective Evaluation Results

Table 4.2 compares the performance of GagaST with different constraints. As described in Section 4.4, we pre-define the note(s) groups and use two syllable-notes assignments: *one-to-one* and *one-to-many*. From results in Table 4.2, we can see,

- The proposed length tag “[LEN\$*i*]” helps to produce lyrics that fit into the predefined note(s) groups. In all cases, less than 30 out of 713 lines produces a longer sentence with ratio less than 0.22; and no short cases.
- We adopt the GagaST w/o constraints except for length tags as our baseline. Compared to which, GagaST with full constraints is able to increase both pitch and rhythm alignments significantly with a fairly slow drop in BLEU.¹¹ It almost doubles the pitch contour alignment score, which affect the intelligibility the most.
- Each constraint applied in decoding process is able to increase the corresponding alignment performance.

¹¹For references, we found three officially translated Disney songs in Mandarin and computes the BLEU among the human translated singable lyrics with the lyrics translation from our dataset, the average BLEU is only 12.3.

Model	Song	<i>sense</i>	<i>style</i>	<i>listenability</i>	<i>intelligibility</i>
GagaST w/o constraints	Song1	3.4	3.0	3.2	3.4
	Song2	3.6	3.9	3.4	3.8
	Song3	3.7	3.6	3.4	3.5
	Song4	3.2	3.0	2.8	3.0
	Song5	3.7	3.6	3.4	3.8
	Average	3.5 ± 0.14	3.4 ± 0.14	3.2 ± 0.12	3.5 ± 0.13
GagaST	Song1	3.5	3.1	3.3	3.5
	Song2	3.4	3.7	3.5	4.0
	Song3	3.2	3.6	3.3	3.6
	Song4	2.9	3.0	3.1	3.5
	Song5	3.4	3.6	3.2	3.9
	Average	3.3 ± 0.15	3.4 ± 0.15	3.3 ± 0.12	3.7 ± 0.13

Table 4.3: Subjective evaluation results for GagaST w/o constraints and GagaST.

4.5.5.2 Subjective Evaluation Results

To examine whether the proposed constraints are able to improve the singability and intelligibility, and to evaluate the quality of translated songs by GagaST, we conduct subjective evaluation and compare the GagaST w/o constraints to fully constrained GagaST in Table 4.3. All songs for subjective evaluation are generated with *one-to-many* assignment. And we compute the confidence intervals for all aspects. Results show that,

- The proposed constraints is able to significantly improve the intelligibility for audience.
- The proposed constraints is able to improve the listening experiences for human with minor significance. The listenability for audience reflects the singability for performer.
- Add constraints do cause a trade-off between the semantics (sense) and other qualities.
- Overall, the annotators are satisfy with the translated songs by the proposed baseline GagaST. All aspects receive an average score around 3.5 out of 5.

The subjective evaluation demonstrates that the proposed alignments, constraints and the acceptable notes transitions for Mandarin (Figure 4.5) are reasonable. They are able to improve the singability and intelligibility. Although there’s a trade-off, due to the lack of singing voice synthesise tools, the subjective evaluation is actually in favour of the sense/style evaluation compared to listenability/intelligibility. We add case studies and post three translated songs by GagaST sung by an amateur singer on <https://gagast.github.io/posts/gagast>.

4.6 Related Work

4.6.1 Verse Generation and Translation

Generating verse text began through rule-based implementations (Milic, 1970) and developed through the next forty years (Gervás, 2000; Levy, 2001; Manurung, 2004; Oliveira, 2012; He et al., 2012; Yan et al., 2013; Zhang and Lapata, 2014; Wang et al., 2016; Ghazvininejad et al., 2016, 2017; Hopkins and Kiela, 2017), incorporating formalisms such as grammars and finite-state machines as reviewed by Oliveira (2017). Poetry translation using these frameworks and statistical machine translation thus offers elegant solutions: Genzel et al. (2010) use phrase-based machine translation technique; while they simply intersect the finite state representation of the meter and rhyme scheme with the synchronous context-free grammar of the translation model. Ghazvininejad et al. (2018) apply the finite-state constraints to neural translation model. However, these representations of the rhythmic and lexical constraints are not flexible enough to encode the real-valued representation of a *song* as required for translation in tonal languages.

4.6.2 Constrained Text Generation

Most natural language generation tasks, including machine translation (Bahdanau et al., 2014; Vaswani et al., 2017; Hassan et al., 2018), dialogue system (Shang et al., 2015; Li et al., 2016) and abstractive summarization (Rush et al., 2015; Paulus et al., 2018), are free text generation. However, there is a need to generate text with some constraints for some special tasks (Lakew et al., 2019; Li et al., 2020; Zou et al., 2021). Hokamp and Liu (2017); Post and Vilar (2018); Hu et al. (2019) attempted to constrain the beam search with dictionary. In the training procedure, Li et al. (2020) added format embedding. Lakew et al. (2019) introduced length tag.

4.6.3 Lyrics Generation

Automatic song translation is a challenging task that involves two fields: machine translation and lyrics generation. As one of the most important tasks in automatic songwriting, lyrics generation has received more attention recently (Malmi et al., 2016; Watanabe et al., 2018; Bao et al., 2019; Lu et al., 2019; Lee et al., 2019; Sheng et al., 2021). Malmi et al. (2016) generated lyrics without melody information; Lee et al. (2019); Bao et al. (2019) attempted to deal with the melody-to-lyrics generation with sequence-to-sequence model. Sheng et al. (2021) use pre-training for melody-to-lyrics generation, but does not take knowledge in the music domain into account.

4.7 Conclusion

This chapter demonstrates the benefits of incorporation rules from human knowledge with data-driven models when the data is scarce. We address automatic

song translation (AST) for tonal languages and the unique challenge of aligning words' tones with melody. And we build the first English-Mandarin AST system – GagaST. Both objective and subjective evaluations demonstrate that GagaST successfully improves the singability and intelligibility of translated songs.

Based on the same methodology, in the future, we can build human-machine collaborated song translation system, which combines human inputs with model outputs on the fly. Song translation is a hard task that requires rich music background knowledge including complex rules that most human translators lack; while competent human translators for prose translations can help to provide much more diverse and faithful translations. One can leverage the diversity of human translations to enrich the searching space and the encoded complex rules by AI systems to ensure singability and intelligibility.

Chapter 5: Proposed Work

In the previous chapter, we address the singable song translations for cross-cultural communications by introducing inductive bias from human knowledge in data-driven model. In this chapter, we propose another novel task that has been overlooked in cross-cultural communications and develop solutions to extend the ability of data-driven models to user needs.

5.1 Adaptive Machine Translation

We propose a novel task—cross-cultural adaptive machine translations (adaptive MT)—to help remove friction in cross-cultural communications. That is, we do not just perform direct translation, but also adapt the entity of interest (*EoI*) in the original sentence, along with the context, into its counterparts in the target culture. In practice, besides direct usage in communication, adaptive MT can also help resolve data bias in developing multilingual models and QA systems, or build connections in cross-cultural document analysis, etc.

5.1.1 Cross-cultural Entity Adaptions

To perform adaptive MT, the first step is to adapt the entity of interest (*EoI*) in the original culture into its counterparts in the target culture. We pretrain Wikipedia entity embeddings for five languages: English, Mandarin, German, French and Spanish; and apply the entity adaption methods proposed by [Peskov et al. \(2021\)](#) to generate cross-cultural entity adaptations.

5.1.2 Complete Cross-cultural Contexts

After obtaining the adaptation for *EoI*, the challenge is to adapt the contexts in the original sentence to fit with the adapted entity of interest (*aEoI*). In the following example,

"How many Grammy Awards has **Beyoncé** won?"

with “Beyoncé” as the *EoI*. Suppose the adapted entity in China is “G.E.M. (Deng Ziqi)”, if we merely adapt the *EoI*, the question would be,

"How many Grammy Awards has **G.E.M. (Deng Ziqi)** won?"

which could be a valid question but is not adapted to the target Chinese culture, since Grammy Awards are presented by the Recording Academy of the United States. A better adapted question should be,

"How many Golden Melody Awards has G.E.M. (Deng Ziqi) won?"

where the Golden Melody Award is a prestigious music award in China, and G.E.M. is a popular Chinese singer.

We propose to address this issue by first running name entity recognition algorithms through the sentences to be adapted and mask out all the named entities in the original sentences. Then we finetune a cross-lingual pretrained language model (Section 2.3.4) and train an entity completion model based on provided contexts. And we use the Wikipedia page of the *EoI* as the context.

5.1.3 Over Generation and Fact Verification Model

After obtaining the adapted sentences where both the *EoI* and its context entities have been adapted into the target culture, it is vital to verify whether the generated sentences are valid in facts. For example, for the following example,

"What film did Beyoncé appear in with Mike Myers?"

and the adapted sentence in Chinese is,

"What film did G.E.M. (Deng Ziqi) appear in with Huang Bo?"

The generated sentences are grammatically correct and "Huang Bo" is indeed a valid adaption for "Mike Myers" (both of them are famous actors). However, this sentence is not valid in fact, since "G.E.M. (Deng Ziqi)" has never been in a movie with "Huang Bo", nor with anyone else. To successfully adapt the original question to Chinese culture, we should find a female singer who has been in a movie with a famous actor. A better adaptation would be,

"What film did Faye Wong appear in with Tony Leung?"

This is a valid adaption in both grammar and facts, and the answer to this adapted question is "Chungking Express".

We propose to over-generate adapted sentences for each source sentence with various choices of *aEoI* and adapted contexts. Then we train a classifier for verifying the facts and remove the invalid adapted sentences.

After filtering the invalid questions, we run traditional machine translation models on the valid adapted sentences and achieve adaptive machine translation.

5.1.4 Applications and Analysis

In this section, we propose to discuss the application of Adaptive MT, such as resolving data bias in developing multilingual models and QA systems, or building connections in cross-cultural document analysis, etc.

Chapter 6: Conclusion and Timeline

In this proposal, we emphasize the importance of human interpretability and human knowledge in the era of data-driven models. We focus on enhancing the interpretability of representations learned by these models, and propose human evaluation metrics to expose the discrepancies between computer-centric task performance and human interpretability of sense-specific embeddings. We improve the interpretability of sense embeddings by differentiable hard attention techniques. Enhancing the interpretability of representation learning allows humans to understand and utilize the learned embeddings more effectively and in a wider range of applications. Furthermore, we integrate human expertise in the form of rules into a song lyrics translation system for tonal languages, guiding the generation of translated lyrics. We explore the often-overlooked musical constraints in the generation of song lyrics for tonal languages and distill rules from musicians and linguists. By introducing inductive bias from human experts and leveraging the power of pretrained large language models, our system is capable of translating song lyrics even in the absence of parallel data. We propose both objective and subjective evaluation metrics to examine whether the translated lyrics meet user expectations in terms of singability and intelligibility. The lyrics generated by our system maintain faithfulness to the original meaning and align with human expectations. This work underscores the benefits of integrating human expertise with machine-learned knowledge (via pretraining) in language generation tasks, particularly when parallel data is scarce. In the proposed work, we leverage the rich semantics learned by the data-driven models and retrieve what the model learns, then we generalize that knowledge to low-resource tasks with human design.

6.1 Timeline

- **2024.5** Proposal
- **2024.6** Submit paper on Adaptive Machine Translation to ARR on June 15th.
- **2025.5** Thesis defense

For the proposed work, we have built the whole adaptation and translation pipeline. And we present a more detailed timeline to complete this submission as follows,

Projects	Timeline	Paper Writing	Timeline
Build evaluation pipeline	April 25	Create structure + abstract	April 25-28
Evaluate and compare pipeline	May 1	Introduction	May 5
Build/Adjust human eval pipeline	May 7	Background	May 12
Run human evaluation	May 14	Model and Experiments	May 20
Summarize results	May 21	Finish a complete paper	May 27

Table 6.1: Adaptation MT paper timeline

Appendix A: Reading List

A.1 Representation Learning

1. Charles Egerton Osgood, George J Suci, and Percy H Tannenbaum. 1957. *The measurement of meaning*. 47. University of Illinois press
2. Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational linguistics*, 24(1)
3. Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828
4. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*
5. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems*
6. Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics*
7. Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*
8. Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of Empirical Methods in Natural Language Processing*
9. Simon Šuster, Ivan Titov, and Gertjan van Noord. 2016. Bilingual learning of multi-sense embeddings with discrete autoencoders. In *Conference of the North American Chapter of the Association for Computational Linguistics*

10. Guang-He Lee and Yun-Nung Chen. 2017. Muse: Modularizing unsupervised sense embeddings. In *Proceedings of Empirical Methods in Natural Language Processing*

A.2 Pretrained Language Model

1. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*
2. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Proceedings of Advances in Neural Information Processing Systems*
3. Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the Association for Computational Linguistics*
4. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Association for Computational Linguistics*
5. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*
6. Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training
7. Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the Association for Computational Linguistics*
8. Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Proceedings of Advances in Neural Information Processing Systems*
9. Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*
10. Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Proceedings of Advances in Neural Information Processing Systems*

A.3 Human Knowledge Controlled Generation

1. Louis T. Milic. 1970. The possible usefulness of poetry generation. In *Symposium on the Uses of Computers in Literary Research*
2. Hugo Gonalo Oliveira. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *INLG*
3. Dmitriy Genzel, Jakob Uszkoreit, and Franz Och. 2010. “poetic” statistical machine translation: Rhyme and meter. In *Proceedings of Empirical Methods in Natural Language Processing*
4. Marjan Ghazvininejad, Yejin Choi, and Kevin Knight. 2018. Neural poetry translation. In *Conference of the North American Chapter of the Association for Computational Linguistics*
5. Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the Association for Computational Linguistics*
6. Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Conference of the North American Chapter of the Association for Computational Linguistics*
7. J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Conference of the North American Chapter of the Association for Computational Linguistics*
8. Surafel Melaku Lakew, Mattia Antonino Di Gangi, and Marcello Federico. 2019. Controlling the output length of neural machine translation. In *16th International Workshop on Spoken Language Translation*
9. Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. 2020. Rigid formats controlled text generation. In *Proceedings of the Association for Computational Linguistics*
10. Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. 2021. Controllable generation from pre-trained language models via inverse prompting. *arXiv preprint arXiv:2103.10685*

Bibliography

- Ben Athiwaratkun, Andrew Wilson, and Anima Anandkumar. 2018. Probabilistic fasttext for multi-sense word embeddings. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Hangbo Bao, Shaohan Huang, Furu Wei, Lei Cui, Yu Wu, Chuanqi Tan, Songhao Piao, and Ming Zhou. 2019. Neural melody composition from lyrics. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 499–511. Springer.
- Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2016. Breaking sticks and ambiguities with adaptive skip-gram. In *Proceedings of Artificial Intelligence and Statistics*.
- Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3.
- Samuel Brody and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of the European Chapter of the Association for Computational Linguistics*.
- Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Frederick Jelinek, John Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of Advances in Neural Information Processing Systems*.
- Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49.
- Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*.
- Henry Chang. 2019. Visualizing ELMo Contextual Vectors.
- Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Proceedings of Advances in Neural Information Processing Systems*.
- Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. 2015. Improving distributed representation of word sense via wordnet gloss composition and context clustering. In *Proceedings of the Association for Computational Linguistics*, pages 15–20. Association for Computational Linguistics.
- Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of Empirical Methods in Natural Language Processing*. Citeseer.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Kenneth Church and Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of the Association for Computational Linguistics*.
- Ido Dagan, Shaul Marcus, and Shaul Markovitch. 1993. Contextual word similarity and estimation from sparse data. In *Proceedings of the Association for Computational Linguistics*.
- Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Association for Computational Linguistics*.
- Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv*.
- Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2009. Investigations on word senses and word usages. In *Proceedings of the Association for Computational Linguistics*.
- Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2013. Measuring word meaning in context. *Computational Linguistics*, 39(3):511–554.
- Allyson Ettinger, Philip Resnik, and Marine Carpuat. 2016. Retrofitting sense-specific word vectors using parallel text. In *Conference of the North American Chapter of the Association for Computational Linguistics*.
- Shi Feng and Jordan Boyd-Graber. 2019. What AI can do for me: Evaluating machine learning interpretations in cooperative play. In *International Conference on Intelligent User Interfaces*.
- Figure Eight. 2018. How to Calculate a Confidence Score.
- Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppín. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1).
- John Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, pages 10–32.
- Johan Franzon. 2008. Choices in song translation: Singability in print, subtitles and sung performance. *The Translator*, 14(2):373–399.
- William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*.
- Dmitriy Genzel, Jakob Uszkoreit, and Franz Och. 2010. “poetic” statistical machine translation: Rhyme and meter. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Pablo Gervás. 2000. Wasp: Evaluation of different strategies for the automatic generation of spanish verse. In *Proceedings of the AISB-00 symposium on creative & cultural aspects of AI*.
- Marjan Ghazvininejad, Yejin Choi, and Kevin Knight. 2018. Neural poetry translation. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

- Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. 2016. Generating topical poetry. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an interactive poetry generation system. In *Proceedings of the Association for Computational Linguistics*.
- José Manuel Gómez-Pérez and Ronald Denaux. 2019. Vecsigrafo: Corpus-based word-concept embeddings—bridging the statistic-symbolic representational gap in natural language processing. *Semantic Web—Interoperability, Usability, Applicability*, 10(6).
- Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
- Erica Greene, Tugba Bodrumlu, and Kevin Knight. 2010. Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor OK Li. 2018. Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368*.
- Emil Julius Gumbel and Julius Lieblein. 1954. Statistical theory of extreme values and some practical applications: a series of lectures.
- Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning sense-specific word embeddings by exploiting bilingual resources. In *Proceedings of International Conference on Computational Linguistics*.
- Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*.
- Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Knowledge Discovery and Data Mining*.
- William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016a. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the Association for Computational Linguistics*.
- William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016b. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the Association for Computational Linguistics*.
- Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

- Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William D. Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *ArXiv*, abs/1803.05567.
- Jing He, Ming Zhou, and Long Jiang. 2012. Generating chinese classical poems with statistical machine translation models. In *Association for the Advancement of Artificial Intelligence*.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4).
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Douglas R Hofstadter. 1997. *Le ton beau de Marot: In praise of the music of language*. Basic Books New York.
- Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the Association for Computational Linguistics*.
- Jack Hopkins and Douwe Kiela. 2017. Automatically generating rhythmic verse with neural networks. In *Proceedings of the Association for Computational Linguistics*.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the Association for Computational Linguistics*.
- Fangzhou Hu. 2017. Lexical tones in mandarin sung words: A phonetic and psycholinguistic investigation. Master’s thesis, Shanghai International Studies University.
- J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Conference of the North American Chapter of the Association for Computational Linguistics*.
- Yuening Hu, Jordan Boyd-Graber, and Brianna Satinoff. 2011. Interactive topic modeling. In *Proceedings of the Association for Computational Linguistics*.
- Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the Association for Computational Linguistics*.

- Ignacio Iacobacci, Taher Mohammad Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the Association for Computational Linguistics*, pages 95–105. Association for Computational Linguistics.
- Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.
- Sujay Kumar Jauhar, Chris Dyer, and Eduard Hovy. 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In *Conference of the North American Chapter of the Association for Computational Linguistics*.
- Thorsten Joachims. 2005. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings*, pages 137–142.
- Martin Joos. 1950. Description of language design. *The Journal of the Acoustical Society of America*, 22(6):701–707.
- Dan Jurafsky and James H. Martin. 2000. *Speech & language processing*. Pearson Education India.
- David Jurgens and Ioannis Klapaftis. 2015. Semeval-2013 task 13: Word sense induction for graded and non-graded senses. In *Proceedings of the Workshop on Semantic Evaluation*.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Daisuke Kawahara, Daniel Peterson, Octavian Popescu, and Martha Palmer. 2014. Inducing example-based semantic frames from a massive amount of verb uses. In *Proceedings of the European Chapter of the Association for Computational Linguistics*.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Proceedings of Empirical Methods in Natural Language Processing*.
- Stephen C Kleene et al. 1956. Representation of events in nerve nets and finite automata. *Automata studies*, 34:3–41.
- Philipp Koehn, Franz J Och, and Daniel Marcu. 2003. Statistical phrase-based translation. Technical report, University of Southern California Marina Del Rey Information Sciences Inst.

- Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the International Conference of Machine Learning*.
- Igor Labutov and Hod Lipson. 2013. Re-embedding words. In *Proceedings of the Association for Computational Linguistics*.
- Surafel Melaku Lakew, Mattia Antonino Di Gangi, and Marcello Federico. 2019. Controlling the output length of neural machine translation. In *16th International Workshop on Spoken Language Translation*.
- Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pre-training. In *Proceedings of Advances in Neural Information Processing Systems*.
- Claudia Leacock, George A. Miller, and Martin Chodorow. 1998. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Guang-He Lee and Yun-Nung Chen. 2017. Muse: Modularizing unsupervised sense embeddings. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Hsin-Pei Lee, Jhih-Sheng Fang, and Wei-Yun Ma. 2019. icomposer: An automatic songwriting system for chinese popular music. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 84–88.
- Robert P Levy. 2001. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proceedings of the ICCBR-01 Workshop on Creative Systems*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the Association for Computational Linguistics*.
- Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. 2016. A persona-based neural conversation model. *ArXiv*, abs/1603.06155.
- Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? *arXiv preprint arXiv:1506.01070*.

- Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. 2020. Rigid formats controlled text generation. In *Proceedings of the Association for Computational Linguistics*.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Peter Low. 2003. Singable translations of songs. *Perspectives: Studies in Translatology*, 11(2):87–103.
- Xu Lu, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. A syllable-structured, contextually-based conditionally generation of chinese lyrics. In *Pacific Rim International Conference on Artificial Intelligence*, pages 257–265. Springer.
- Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Conference on Computational Natural Language Learning*.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov).
- Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.
- Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis. 2016. Dopelearning: A computational approach to rap lyrics generation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 195–204.
- Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. 2017. Embedding words and senses together via joint knowledge-enhanced training. In *Conference on Computational Natural Language Learning*.
- Hisar Manurung. 2004. *An evolutionary algorithm approach to poetry generation*. Ph.D. thesis, University of Edinburgh.
- Diana McCarthy, Marianna Apidianaki, and Katrin Erk. 2016. Word sense clustering and clusterability. *Computational Linguistics*, 42(2):245–275.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems*.
- Louis T. Milic. 1970. The possible usefulness of poetry generation. In *Symposium on the Uses of Computers in Literary Research*.
- George Miller and Christiane Fellbaum. 1998. *Wordnet: An electronic lexical database*. MIT Press Cambridge.
- George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6.
- George A Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of the workshop on Human Language Technology*.
- Tristan Miller, Christian Hempelmann, and Iryna Gurevych. 2017. SemEval-2017 Task 7: Detection and interpretation of English puns. In *Proceedings of the Workshop on Semantic Evaluation*.
- Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the International Conference of Machine Learning*.
- Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Dai Quoc Nguyen, Dat Quoc Nguyen, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. 2017. A mixture model for learning multi-sense word embeddings. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics*.
- Yoshiki Niwa and Yoshihiko Nitta. 1994. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of International Conference on Computational Linguistics*.
- Thanapon Noraset, Chen Liang, Lawrence A Birnbaum, and Douglas C Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Association for the Advancement of Artificial Intelligence*.
- Hugo Gonalo Oliveira. 2012. Poetryme: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence*, 1:21.
- Hugo Gonalo Oliveira. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *INLG*.
- OpenAI*. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

- P. Orbanz and Y. W. Teh. 2010. *Bayesian Nonparametric Models*. Springer.
- Charles Egerton Osgood, George J Suci, and Percy H Tannenbaum. 1957. *The measurement of meaning*. 47. University of Illinois press.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Proceedings of Advances in Neural Information Processing Systems*.
- Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. *ArXiv*, abs/1705.04304.
- Maria Pelevina, Nikolay Arefiev, Chris Biemann, and Alexander Panchenko. 2016. Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Christian S Perone, Roberto Silveira, and Thomas S Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*.
- Denis Peskov, Viktor Hangya, Jordan Boyd-Graber, and Alexander Fraser. 2021. Adapting entities across languages and cultures. In *emnlp*.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics*.
- Mohammad Taher Pilehvar and Jose Camacho-Collados. 2018. Wic: 10,000 example pairs for evaluating context-sensitive representations. *arXiv preprint arXiv:1808.09121*.
- Mohammad Taher Pilehvar and Nigel Collier. 2016. De-conflated semantic representations. In *Proceedings of Empirical Methods in Natural Language Processing*.

- Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Conference of the North American Chapter of the Association for Computational Linguistics*.
- Amruta Purandare and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Conference on Computational Natural Language Learning*.
- Lin Qiu, Kewei Tu, and Yong Yu. 2016. Context-dependent sense embedding. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Lawrence R Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.
- Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the World Wide Web Conference*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.
- ITU Rec. 1994. P. 85. a method for subjective performance assessment of the quality of speech voice output devices. *International Telecommunication Union, Geneva*.
- Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Conference of the North American Chapter of the Association for Computational Linguistics*.
- Sascha Rothe and Hinrich Schütze. 2017. Autoextend: Combining word embeddings with semantic resources. *Computational Linguistics*, 43(3).
- Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10).
- Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*.
- Devendra Singh Sachan, Mike Lewis, Dani Yogatama, Luke Zettlemoyer, Joelle Pineau, and Manzil Zaheer. 2022. Questions are all you need to train a dense passage retriever. *Transactions of the Association for Computational Linguistics*.
- Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

- Murray Henry Schellenberg. 2013. *The realization of tone in singing in Cantonese and Mandarin*. Ph.D. thesis, University of British Columbia.
- Tobias Schnabel, Igor Labutov, David M. Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Marius Schneider. 1961. Tone and tune in west african music. *Ethnomusicology*, 5(3):204–215.
- Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational linguistics*, 24(1).
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the Association for Computational Linguistics*.
- Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of the International Conference on Learning Representations*.
- Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *ACL*.
- Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813.
- Zhonghao Sheng, Kaitao Song, Xu Tan, Yi Ren, Wei Ye, Shikun Zhang, and Tao Qin. 2021. Songmass: Automatic song writing with pre-training and alignment constraint. In *Proceedings of the Association for Computational Linguistics*.
- Simon Šuster, Ivan Titov, and Gertjan van Noord. 2016. Bilingual learning of multi-sense embeddings with discrete autoencoders. In *Conference of the North American Chapter of the Association for Computational Linguistics*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Proceedings of Advances in Neural Information Processing Systems*.
- Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. 2014. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of International Conference on Computational Linguistics*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Proceedings of Advances in Neural Information Processing Systems*.

- Chong Wang, John Paisley, and David M. Blei. 2011. Online variational inference for the hierarchical Dirichlet process. In *Proceedings of Artificial Intelligence and Statistics*.
- Qixin Wang, Tianyi Luo, Dong Wang, and Chao Xing. 2016. Chinese song iambics generation with neural attention-based model. In *International Joint Conference on Artificial Intelligence*.
- Kento Watanabe, Yuichiroh Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano. 2018. A melody-conditioned lyrics language model. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 163–172.
- Lian Hee Wee. 2007. Unraveling the relation between mandarin tones and musical melody. *Journal of Chinese Linguistics*, 35(1):128.
- Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.
- Zhaohui Wu and C Lee Giles. 2015. Sense-aware semantic analysis: A multi-prototype word representation model using wikipedia. In *Association for the Advancement of Artificial Intelligence*, pages 2188–2194. Citeseer.
- Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Conference of the North American Chapter of the Association for Computational Linguistics*.
- Yi Xu. 1997. Contextual tonal variations in mandarin. *Journal of phonetics*, 25(1):61–83.
- Rui Yan, Han Jiang, Mirella Lapata, Shou-De Lin, Xueqiang Lv, and Xiaoming Li. 2013. I, poet: automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *International Joint Conference on Artificial Intelligence*.
- Dongqiang Yang and David Martin Powers. 2006. *Verb similarity on the taxonomy of WordNet*. Masaryk University.
- Yang Yinliu, Sun Congyin, and Wu Junda. 1983. *Language and Music*. People’s Music Publishing House.
- Moira Yip. 2002. *Tone*. Cambridge University Press.
- Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of Empirical Methods in Natural Language Processing*.

- Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. 2021. Controllable generation from pre-trained language models via inverse prompting. *arXiv preprint arXiv:2103.10685*.
- Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. 2015. Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian document computing symposium*, pages 1–8.