論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	A Study of Non-standardWord Usage on Social Media
著者(和文)	青木竜哉
Author(English)	Tatsuya Aoki
出典(和文)	学位:博士(工学), 学位授与機関:東京科学大学, 報告番号:甲第283号, 授与年月日:2025年3月26日, 学位の種別:課程博士, 審査員:奥村 学,中山 実,鈴木 賢治,篠﨑 隆宏,船越 孝太郎,高村 大也
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Institute of Science Tokyo, Report number:甲第283号, Conferred date:2025/3/26, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Doctoral Thesis

Academic Year 2025

A Study of Non-standard Word Usage on Social Media

Tatsuya Aoki

Department of Information and Communications Engineering
School of Engineering

Institute of Science Tokyo

Supervisor Manabu Okumura, Professor

Abstract

User-generated texts contain not only non-standard words, such as b4 for before, but unusual word usages, such as catfish for a person who uses a fake identity online, which requires knowledge about the words to handle such cases in natural language processing.

This dissertation studies these unconventional word usages in social media contexts to gain a better understanding of informal language and to advance natural language processing systems in such settings. To support this effort, we introduce a new dataset that captures non-standard word usages in both English and Japanese, and we present a neural model for detecting these usages from large-scale text streams. To deal with the lack of training data for this task, we propose a method for synthetically generating pseudo non-standard examples from a corpus, which enables us to train the model without manually-annotated training data and for any arbitrary language. Experimental results on X and Reddit datasets show that our proposed method achieves better performance than existing methods, and is effective across different languages.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Social media text often contains slang and other non-standard language usages, posing problems for natural language processing (NLP) (Eisenstein, 2013). Much research has addressed this issue at the word-type level, where a word-type refers to a unique word within a text corpus, tackling expressions such as *b4* to mean *before* or *soz* to mean *sorry* (Aw et al., 2006; Han and Baldwin, 2011; Han et al., 2013; Li and Liu, 2015; Barteld, 2017; Stewart and Eisenstein, 2018; Kulkarni and Wang, 2018; Lourentzou et al., 2019; Cho and Kim, 2021; Sun et al., 2022). However, to fully understand informal language use, it is also essential to examine word-token level usages, where a word-token refers to a specific instance of a word as it appears in a particular context. For example, in the sentence *The person turned out to be a catfish from an online dating app*, *catfish* means "a person who uses a fake online profile", diverging from its original definition of an aquatic animal (Magdy et al., 2022). This shift highlights how a single dictionary-listed word can evolve a context-specific sense in online conversations.

This dissertation investigates the word-token level usage of dictionary-listed words on social media that deviates from standard conventions. We refer to these deviations as **non-standard word usages**: non-standard extensions of existing word senses that diverge from their conventional or literal meanings. Unlike word-type level slangs, jargons or new word occurrences, such as *b4* or *soz*, non-standard word usages emerge when an established word takes on an additional meaning or function within particular social or cultural contexts, often exhibiting creative or context-specific nuances.

Figure 1.1 further illustrates our primary research objective, highlighting the distinction between In-Vocabulary (IV) and Out-of-Vocabulary (OOV) words. Our central

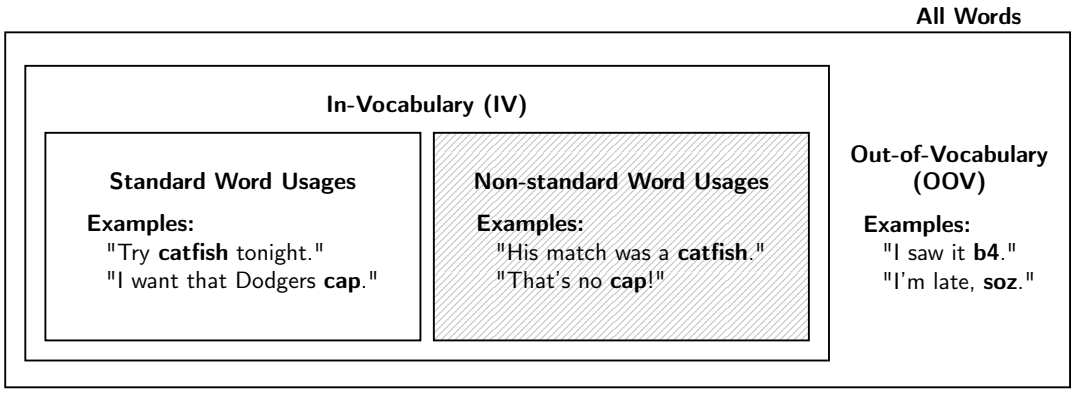

Figure 1.1: Illustration of our primary research focus (striped area): In-Vocabulary words used as a non-standard usage. For example, **catfish** denotes a type of fish but can also mean someone misrepresenting themselves online, and **cap** typically means "hat" but can also mean "lie" in slang. Out-of-Vocabulary words, such as **b4** or **soz**, fall beyond the scope of this thesis.

focus is on handling known IV words that exhibit non-standard or evolving meanings, such as *catfish* or *cap* in Figure 1.1, rather than completely new terms that fall into the OOV category, such as *b4* or *soz* in the figure. This distinction is crucial because a single IV term may appear in both standard and non-standard senses, creating ambiguities that a robust NLP system must resolve. Consequently, the figure raises a fundamental research question: **can we reliably distinguish non-standard word usages to advance NLP systems for informal language understanding?** Before we delve into the research topic, we first examine why these emergent word usages warrant deeper investigation, as following sections will clarify the broader complexities they impose on NLP systems.

Non-standard word usages pose a unique challenge for NLP systems, owing to their evolving and context-dependent nature. They are frequently employed to convey sensitive content in ways that evade censorship (Huang et al., 2013; Zhang et al., 2015; Steen et al., 2023), necessitating systematic detection to prevent online harms (Fillies and Paschke, 2024). Non-standard word usages also exist not only in English but also in other languages (Sboev, 2016), which causes an issue in downstream NLP tasks such as machine translation. As an example, Figure 1.2 demonstrates how a machine translation model processes an input sentence containing a non-standard word usage. In this example, as of December 2024, Google Translate provides a literal translation of the Japanese word "鯖 (mackerel)", failing to capture its meaning of the slang sense "サーバー (computer server)", which arises from its phonetic similarity. Hence, the correct translation accounting for the non-standard word usage would be "Twitter's computer

Figure 1.2: Result from Google Translate for non-standard word usage.

servers have gone down." This example illustrates a broader problem: most models lack the lexical and contextual resources needed to accurately handle these non-standard word usages.

The difficulty in resolving non-standard word usages within NLP systems is primarily due to the lack of comprehensive resources documenting these usages, causing models to misinterpret their alternative meanings. Recent research shows that knowledge of word usages plays a crucial role in improving the quality of downstream NLP tasks including machine translation (Hangya et al., 2021; Campolungo et al., 2022), language modeling (Zhang et al., 2020) or question answering (Fang et al., 2022), implying that non-standard language use can be handled within the same framework. Nevertheless, even though linguistic resources such as BabelNet (Navigli and Ponzetto, 2010) provide extensive semantic networks, they omit many slang senses, creating further obstacles for training models capable of recognizing non-standard word usages.

The systematic collection of non-standard word usages is therefore essential for improving NLP systems, and the detection of these usages is particularly crucial. This is because, as a first step in populating language resources with these expressions, we must systematically extract them from large-scale texts. However, there are three major challenges that must be addressed before a reliable detection method can be developed.

1. **Lack of annotated data for non-standard word usages.**
   This scarcity prevents supervised machine learning models from being trained effectively, thereby posing additional challenges. Recent efforts have attempted to build datasets for slang and non-standard usages. For instance, Sun et al. (2024) constructed a slang dataset from English movie subtitles. Despite ongoing advances, there remains a lack of training datasets covering a wide range of domains, causing semi-supervised methods to struggle with cross-domain knowledge transfer (Ruder and Plank, 2018).

2. **The need for multilingual systems.**
   Since non-standard word usages are not confined to English but also arise in other languages (Sboev, 2016), it is essential to develop a scalable, language-independent approach to acquiring and understanding them. Given their multilingual nature, failing to address these variations can limit the generalizability of NLP models on social media data. Consequently, cross-lingual analysis of detection methods becomes vital for addressing such usages across diverse linguistic contexts.

3. **The necessity of a lightweight approach.**
   Given the massive volume of social media data, relying solely on computationally intensive methods, such as the one based on an in-house large language model (LLM), can be impractical. Likewise, using third-party LLMs, such as ChatGPT (OpenAI, 2023), would be prohibitively expensive for large-scale deployment. Instead, more efficient, lightweight solutions are required to analyze and monitor evolving non-standard word usages at scale, without incurring excessive computational or financial costs.

In this dissertation, we study non-standard word usages both in English and Japanese to deepen our understanding of informal languages and to advance NLP systems accordingly. We address the aforementioned challenges in non-standard word usage detection by developing the following two language-independent, annotation-free methods:

- **Word Embedding-based Method.**
  This approach incorporates word embeddings to capture semantic similarities between words, enabling the identification of non-standard usages by analyzing deviations from standard patterns. We employ Skip-gram (Mikolov et al., 2013) for learning word representations, which can be done in a language-independent manner and is a lightweight solution. To train these embeddings, this method utilizes a balanced set of usage examples so that each word usage is represented proportionally, allowing the model to effectively distinguish non-standard from standard usages. Furthermore, by manipulating the learning algorithm of Skip-gram with negative sampling and comparing dot products between target and context words, the approach can detect which words are used in a non-standard way, leveraging the common-sense knowledge encapsulated in the trained word embeddings.

- **Masked Language Model-based Method.**
  This approach leverages masked language models (MLMs) to evaluate a word usage within a broader context, enabling more accurate detection of non-standard expressions than the word embedding-based method. We employ RoBERTa (Liu et al., 2019), a widely used MLM in modern NLP, which is available in over 100 languages making them well-suited for multilingual scenarios. To eliminate the need for large annotated corpora, this method employs pseudo-label training (Bergsma et al., 2008; Poon et al., 2009) by generating synthetic datasets for non-standard word usage detection, which are then applied in real-world settings. A classifier layer is built on top of the MLM components and is fine-tuned using these pseudo-labeled corpora, allowing the system to predict whether a given word usage is non-standard. This design remains sufficiently lightweight to be feasibly applied to large-scale, real-world text streams, thereby addressing the need for scalable, multilingual solutions.

In addition, we introduce a new dataset designed to advance research in this area. This dataset comprises data from English Reddit[1] and Japanese X[2] (formerly Twitter), allowing us to examine the robustness of our methods across different linguistic settings. Unlike Sun et al. (2024), which focused on slang expressions in movie subtitles, our dataset draws upon social media texts, thereby capturing the highly dynamic and evolving nature of real-world, user-generated languages. As such, it serves as a more representative benchmark for studying non-standard word usages in authentic online contexts.

In our experiments, we first concentrate on the task of word usage classification, where the objective is to determine whether a given word in a sentence is used in a non-standard way. We evaluate this classification task using a Japanese X dataset, a newly created expert-annotated corpus tailored for word usage classification. We then extend our approach to a non-annotated English Reddit dataset, discovering non-standard usages with evaluations conducted by crowd-workers. The results on both the Japanese X dataset and the English Reddit dataset demonstrate that our proposed model can effectively detect non-standard usages across different languages and social media platforms.

[1] https://www.reddit.com
[2] https://x.com

1.2 Contributions

Non-standard word usages have become increasingly pivotal in NLP, particularly as social media content diversifies and expands. This thesis addresses the complexities of detecting and analyzing non-standard word usages in multilingual social media data. The primary contributions of this work are summarized as follows:

- **Methods for Non-standard Word Usage Detection on Social Media.**
  We introduce two distinct approaches that do not rely on specialized non-standard usage annotations:
  - A *word embedding-based approach* that employs Skip-gram and balanced corpora to learn semantic representations in a lightweight, language-agnostic manner. By comparing deviations from common-sense norms embedded within these learned vectors, this approach effectively identifies non-standard usages without extensive manual labeling.
  - A *masked language model (MLM)-based approach* that employs RoBERTa and a word usage classifier layer enhanced with pseudo-label training.
    RoBERTa is available in multiple languages, making this method inherently language-agnostic and suitable for diverse linguistic contexts. Synthetic examples are generated from unannotated corpora, enabling more robust detection of non-standard word usages across varying real-world scenarios.
- **Construction of Novel Datasets for Non-standard Word Usages.**
  We curate new datasets for non-standard word usages on English Reddit and Japanese X. By focusing on authentic, user-generated text, these datasets more accurately represent the evolving and informal nature of language on social media.
- **Multilingual Evaluation of Non-standard Word Usages.**
  Our methods are systematically evaluated on both English and Japanese datasets, demonstrating adaptability across languages with distinct linguistic features. Expert-annotated data and crowd-sourced evaluations confirm the scalability and effectiveness of the proposed approaches.

These contributions collectively advance the detection and analysis of non-standard word usages, offering new datasets and innovative methodologies, and a comparative framework for handling informal languages in large-scale, multilingual social media contexts.

1.3 Outline of Thesis

This thesis investigates methods for detecting non-standard word usages and a crucial aspect of understanding informal languages, by developing novel computational approaches. The structure of the thesis is organized as follows:

- **Chapter 2** surveys existing literature on non-standard word usages and reviews two foundational NLP components: representation learning by the Skip-gram model and masked language modeling with RoBERTa.
- **Chapter 3** first creates a dataset of non-standard word usages of X in Japanese, annotated by experts to conduct the analysis of this phenomenon. We then introduce a *word embedding-based method* that detects non-standard usages in an unsupervised manner by examining context-word similarities.
- **Chapter 4** proposes a *masked language model-based method* augmented with pseudo-label training, eliminating the need for extensive annotated corpora. This chapter also introduces an English Reddit dataset annotated by crowd-workers and presents a comprehensive evaluation of both the word embedding-based and masked language model-based methods on Japanese and English datasets.
- **Chapter 5** concludes the thesis by summarizing the main findings, discussing broader implications for NLP, and proposing directions for future work.

Chapter 2

Related Work

In this chapter, we first review the existing literature on the use of informal language in social media and related studies in natural language processing. We then focus on word embedding and masked language modeling, highlighting key considerations for their application to non-standard word usages and examining how these approaches have been utilized in related work.

2.1 Informal Language Use on Social Media

Eisenstein (2013) reported that social media text contains a lot of non-standard usages, including emoticons and abbreviations, which some researchers have attempted to normalize at the lexical level (Han and Baldwin, 2011). There has also been work on predicting the meaning of rare or unknown words (Stewart and Eisenstein, 2018; Yan et al., 2020). One of the related areas of research is slang (Kulkarni and Wang, 2018; Pei et al., 2019; Sun et al., 2022). Particularly, Sun et al. (2024) focus on broad slang as their research subject, examining not only in-vocabulary words and multi-word expressions such as *sugar daddy*, but also other slang terms like *innit*, used to mean "isn't it", or *bruv*, meaning "brother". In our study, we focus solely on the usages of in-vocabulary words appearing in social media texts generated by users, whereas their research mainly deals with subtitles in the movie domain. Therefore, our research differs from that of Sun et al. (2024) in terms of the research subject. Also, the term "slang" can be somewhat ambiguous in this context. Specifically, it may refer to a word itself being considered slang (word level) or to a particular usage being considered slang (token level). In this study, we use the term "non-standard word usage" to avoid this ambiguity, referring only to the latter case.

Elsewhere, researchers have focused on in-vocabulary words that have novel or special uses in a particular domain. Gella et al. (2014) observed that individual Twitter users tend to favor specific senses for particular words, which may differ from user to user. Bamman et al. (2014) proposed a model to learn word representations that reflect geographical differences in language usage. Hamilton et al. (2016) and Yang and Eisenstein (2017) reported that, on social media such as Twitter, words can have opposite sentiment polarities depending on the author or communities that the author belongs. Other research focuses on investigating specific language use across communities. For example, Del Tredici and Fernández (2018) conducted an analysis of language use across different communities, while Lucy and Bamman (2021) attempted to investigate meanings in niche communities.

2.2 Natural Language Processing for Evolving Word Usages

In the following sections, we explore how a range of NLP tasks and methodologies can shed light on the complexities of informal language, newly emerging usages, and domain-specific expressions in user-generated texts.

2.2.1 Word Sense Disambiguation

Word sense disambiguation (WSD) is a fundamental task in NLP (Ide and Véronis, 1998; Navigli, 2009), closely related to our work. WSD systems are typically trained on relatively clean, well-curated text and reference senses from static inventories. However, social media platforms introduce non-standard word usage, including dynamic slang, creative orthography, and rapid sense evolution, that often fall outside typical settings.

As Liu and Liu (2023) reported, non-standard word usages on social media are frequently out-of-distribution compared to models trained on conventional corpora, where they claimed that such usages can lead WSD models to overfit or misinterpret novel senses with high confidence. Moro et al. (2014) proposed Babelfy, a unified approach to WSD and entity linking, which leverages a lexicalized semantic network to generate semantic signatures for concepts and named entities. By treating WSD and entity linking as interconnected tasks, Babelfy offers a more flexible framework that can potentially adapt to the dynamic nature of social media language. Although leveraging

extra-linguistic cues can be beneficial (Barnard et al., 2003), the unpredictable context of social media posts, which often include memes, emojis, or abbreviations, complicates disambiguation. As an alternative, Bejgu et al. (2024) proposed Word Sense Linking, a more flexible approach that moves beyond traditional assumptions of predefined sense candidates and pre-identified ambiguous spans. Their method automatically identifies spans to disambiguate and links them to appropriate senses, but it still relies on a fixed sense inventory and may struggle with completely novel senses, such as emerging social media terms like *catfish*.

2.2.2 Semantic Change Detection

Semantic change detection (SCD) is yet another close research field (Kulkarni et al., 2015; Hamilton et al., 2016; Kutuzov et al., 2018; Hu et al., 2019; Giulianelli et al., 2020; Montariol et al., 2021; Inoue et al., 2022; Nagata et al., 2023). In the field of diachronic semantic change, their focus is on employing word similarity calculations across different time spans (Kulkarni et al., 2015; Hamilton et al., 2016; Giulianelli et al., 2020; Martinc et al., 2020; Nagata et al., 2023) or clustering using topic models (Montariol et al., 2021; Inoue et al., 2022), which differs from our main research objective. Tang et al. (2023) propose a sense-centric method for detecting semantic changes across corpora. Their approach leverages pretrained, static sense information to automatically annotate word occurrences with sense IDs, then compares the resulting sense distributions across corpora using divergence or distance measures. However, as with the WSD approaches discussed above, this method relies on fixed sense information, which may limit its capacity to address non-standard word usages or rapidly evolving language not well-represented in existing sense inventories.

2.2.3 Unknown Sense Detection

Most relevant to this study is research on detecting unknown or novel senses in web corpora. Erk (2006) addressed the problem of unknown word sense detection, which involves identifying corpus occurrences not covered by a given sense inventory, an area where many WSD or SCD methods struggle due to non-standard word usages. Lau et al. (2012) introduced the task of novel sense detection, where they attempted to identify novel word senses via word sense induction through topic modelling, which Cook et al. (2014) extended by leveraging novelty scores for word senses. Building

on these foundations, Lau et al. (2014) introduced a method for learning word sense distributions and detecting unattested senses, particularly useful in domain adaptation. Their approach employs topic models as a proxy for sense, enabling the grouping of tokens that exhibit the same novel sense.

One important distinction, however, is that these studies primarily focus on word-type level processing rather than individual usage. This contrasts with our core research objective, which requires analysis at the word-token level by examining each instance of word usage in context. Moreover, as noted by Li et al. (2016) and Wu et al. (2022), classic topic modeling approaches often face challenges when applied to shorter texts (such as those on X), due to data sparsity and limited word co-occurrence information. This limitation suggests that alternative methods may be more suitable for detecting non-standard word usage in such contexts.

2.3 Word Embedding-based Non-standard Word Usage Detection

2.3.1 Word Embedding

Word embedding is a cornerstone of modern natural language processing (NLP), enabling machines to understand and process human language with greater nuance and accuracy. At their core, word embeddings are dense vector representations of words in a continuous vector space, where semantically similar words are positioned closely together (Mikolov et al., 2013). Unlike traditional one-hot encoding methods, such as Bag-of-Words modeling, which result in high-dimensional and sparse vectors, word embeddings capture intricate semantic relationships in a dense and efficient manner.

The concept of word embeddings is grounded in the distributional hypothesis (Harris, 1954), which posits that words appearing in similar contexts tend to have similar meanings. This principle is operationalized through various models that learn embeddings by analyzing large corpora of text, identifying patterns and co-occurrences of words within predefined contexts.

Mikolov et al. (2013) proposed the Continuous Bag-of-Words (CBOW) and Skip-gram architectures, which demonstrated significant improvements in capturing semantic and syntactic relationships. These neural models leverage shallow neural networks to predict target words from their contexts (or vice versa), effectively learning meaningful

vector representations in the process. They also published a widely used software tool called Word2Vec[1] to learn word embeddings using these two methods. Other models, such as GloVe (Pennington et al., 2014) and FastText (Bojanowski et al., 2016), have further enhanced the quality and applicability of word embeddings. GloVe combines global matrix factorization with local context window methods, while FastText extends the Word2Vec approach by incorporating subword information. This extension allows the model to generate embeddings for out-of-vocabulary words through the aggregation of character n-grams.

In summary, word embeddings provide a foundational framework for representing and manipulating textual data in a continuous vector space, facilitating more sophisticated and effective NLP models. The subsequent subsections dive deeper into specific embedding techniques of Skip-gram model.

2.3.2 Skip-gram with Negative Sampling

The Skip-gram model, introduced by Mikolov et al. (2013), is a widely used neural network-based approach for learning high-quality distributed word representations. It is a key component of the word2vec framework and has significantly influenced subsequent developments in natural language processing and word embedding techniques.

In Skip-gram, given a sequence of words $w_1, w_2, \ldots, w_T$ in the training data and a window size m, the objective is to maximize the following expression:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq i\leq m, i\neq 0}\log p(w_{t+i}|w_t),$$

where $p(w_k|w_t)$, the probability of the context word w_k given the target word w_t, is defined as:

$$p(w_k|w_t) = \frac{\exp(\mathbf{v}^{IN}_{w_t}\cdot\mathbf{v}^{OUT}_{w_k})}{\sum_{w\in W}\exp(\mathbf{v}^{IN}_{w_t}\cdot\mathbf{v}^{OUT}_{w})}.$$

Here, W represents the vocabulary of the training data, and $\mathbf{v}^{IN}$ and $\mathbf{v}^{OUT}$ are the input and output word vectors, respectively. The model predicts context words surrounding a given target word, using these vectors to compute probabilities.

To reduce the computational cost of training, Mikolov et al. (2013) proposed Skip-gram with Negative Sampling (SGNS). SGNS modifies the Skip-gram objective to

[1] https://code.google.com/archive/p/word2vec

maximize the following expression when the target word w_t and a context word w_k appear close to each other:

$$\log \sigma(\mathbf{v}_{w_t}^{IN} \cdot \mathbf{v}_{w_k}^{OUT}) + \sum_{n=1}^{N} \mathbb{E}_{w_n \sim Z(w)} \log \sigma(-\mathbf{v}_{w_t}^{IN} \cdot \mathbf{v}_{w_n}^{OUT}),$$

where σ denotes the sigmoid function, N is the number of negative samples, and $Z(w)$ is a probability distribution used to sample negative examples. In SGNS, when w_t and w_k co-occur, the term $\log \sigma(\mathbf{v}_{w_t}^{IN} \cdot \mathbf{v}_{w_k}^{OUT})$ is maximized, while $\log \sigma(-\mathbf{v}_{w_t}^{IN} \cdot \mathbf{v}_{w_n}^{OUT})$ is minimized for negative samples w_n. This optimization allows SGNS to efficiently learn word vectors that reflect co-occurrence patterns in the corpus.

When measuring word similarity, the input word vectors $\mathbf{v}^{IN}$ are widely used. However, relatively few studies, such as those by Mitra et al. (2016) and Press and Wolf (2017), have effectively utilized the output word vectors $\mathbf{v}^{OUT}$. Levy and Goldberg (2014) demonstrated the equivalence between SGNS and Shifted Positive Pointwise Mutual Information (SPPMI), showing that the use of both $\mathbf{v}^{IN}$ and $\mathbf{v}^{OUT}$ in SGNS is related to leveraging word co-occurrence information encoded in SPPMI. This equivalence implies that the strength of association between a target word and its surrounding words can be effectively captured using these vectors.

Considering the training process of input and output word vectors in SGNS and its equivalence to SPPMI, it becomes clear that calculating word similarity should not solely rely on cosine similarity between input vectors ($\mathbf{v}_{w_t}^{IN}$ and $\mathbf{v}_{w_k}^{IN}$), as commonly done in prior research. Instead, similarity measures that incorporate both input and output vectors, such as $\sigma(\mathbf{v}_{w_t}^{IN} \cdot \mathbf{v}_{w_k}^{OUT})$, should also be considered for a more comprehensive evaluation of word relationships.

In this thesis, SGNS serves as a foundation for developing our word embedding-based approach to identifying non-standard word usage. Its ability to capture semantic similarities is particularly valuable for understanding deviations in word usage, which is a core aspect of our proposed methodology.

2.4 Masked Language Model-based Non-standard Word Usage Detection

2.4.1 Transformer-based Language Modeling

Language models are fundamental components in NLP that assign probabilities to sequences of words, enabling tasks such as text generation, translation, and comprehension. Formally, a language model defines a probability distribution $P(\mathbf{x})$ over a sequence of tokens $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, where:

$$P(\mathbf{x}) = \prod_{i=1}^{n} P(x_i | x_1, x_2, \ldots, x_{i-1}).$$

Vaswani et al. (2017) introduced transformer, a widely used framework in neural network-based language models. Unlike Recurrent Neural Networks (RNNs) (Cho et al., 2014) or Long Short-Term Memory (LSTM) networks, transformer processes all tokens in a sequence simultaneously, thereby enabling efficient parallelization and the capture of long-range dependencies.

Figure 2.1 presents an overview of the transformer architecture. The model consists of multiple layers, each integrating multi-headed self-attention and feed-forward networks. Layer normalization (Ba et al., 2016) stabilizes training and accelerates convergence, while residual connections (He et al., 2016) promote efficient gradient flow, mitigating the risk of vanishing gradients.

An autoregressive language model, including the transformer, is trained to predict the next token in a sequence given all previously observed tokens. Formally, given a tokenized sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, the training objective is to minimize the negative log-likelihood of the observed tokens:

$$\mathcal{L}_{LM} = -\sum_{i=1}^{n} \log p(x_i \mid x_{<i}; \theta),$$

where θ denotes the parameters of the model, and $p(x_i \mid x_{<i})$ is the model's conditional probability of the token x_i given all preceding tokens $x_{<i} = (x_1, \ldots, x_{i-1})$. Cross-entropy loss provides a practical way to evaluate how closely the predicted probabilities of the model match the actual observed tokens. By minimizing this loss, the model learns to assign higher probability to the correct token at each step, thereby improving its accuracy

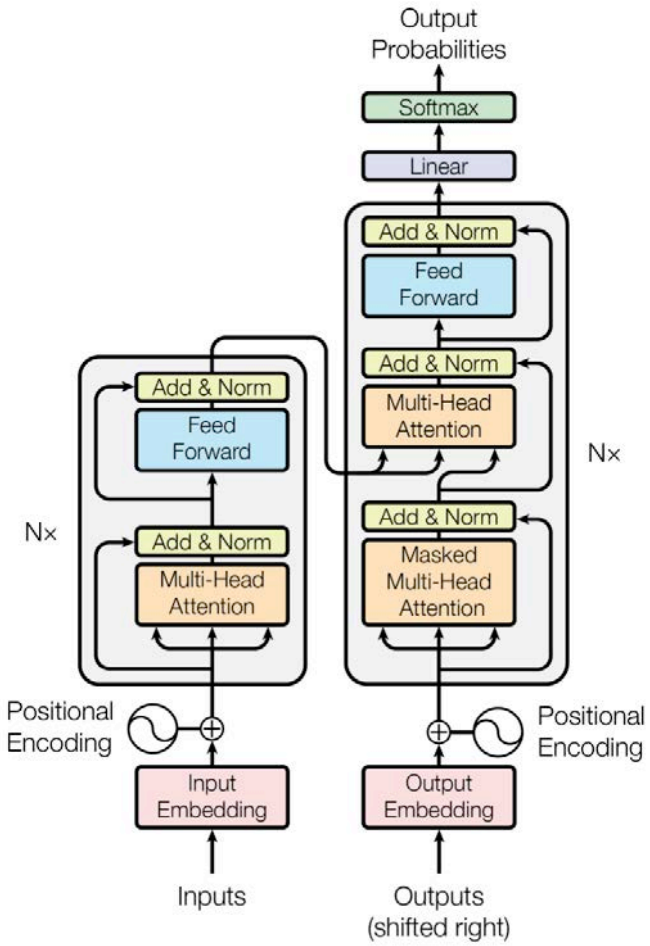

Figure 2.1: The architecture of transformer. Source: (Vaswani et al., 2017)

in predicting the next token.

2.4.2 RoBERTa-based Masked Language Modeling

Masked Language Modeling (MLM) has become a popular way to pre-train the language models for further inference or downstream task, particularly following the introduction of models such as BERT (Devlin et al., 2019). MLM is widely used to learn contextualized representations (Peters et al., 2018), leveraging hidden representations from masked token prediction to capture deep bidirectional context, which can be effectively applied in various NLP downstream tasks (Yang et al., 2023; Wettig et al., 2023). Unlike traditional left-to-right or autoregressive language models, MLM is trained over partially masked input sentences, where a subset of tokens is replaced with a special [MASK] token. The

model then learns to predict these masked tokens based on their context, enabling the capture of bidirectional dependencies within text. Formally, given a sequence of tokens $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, a subset of tokens is randomly selected and replaced with [MASK], yielding the masked sequence $\mathbf{x}_{\text{masked}}$. The MLM objective is to maximize the likelihood of the original tokens given this masked sequence:

$$\mathcal{L}_{\text{MLM}} = -\sum_{x_i \in \mathbf{x}_{\text{masked}}} \log p(x_i \mid \mathbf{x}_{\text{masked}}, \theta),$$

where θ denotes the model parameters. This formulation encourages the model to learn context-aware representations by leveraging both left and right contexts, thereby enriching its understanding of language structure and meaning.

Liu et al. (2019) introduced RoBERTa, the bidirectional transformer-based MLM architecture, that set new benchmarks across a wide range of NLP tasks, including sentiment analysis, question answering, and text classification. RoBERTa employs a dynamic masking strategy during training, generating multiple masked versions of the same text. This approach involves randomly selecting and masking different tokens each time a sequence is fed to the model. The masking process in RoBERTa follows these steps:

- 80% of the time: Replace with the [MASK] token.
- 10% of the time: Replace with a random token.
- 10% of the time: Leave unchanged.

This varied approach helps the model learn more robust representations and prevents it from relying too heavily on the [MASK] token or specific masked patterns. By dynamically changing the masked tokens in each epoch, RoBERTa ensures that the model encounters a diverse range of contexts for each word. The remainder of this section details how the RoBERTa model contextualizes masked tokens within a given input.

Figure 2.2 illustrates the essential components and workflow of RoBERTa for masked language modeling. Formally, let the tokenized sequence be $\mathbf{x} = (x_1, x_2, \ldots, x_i, \ldots, x_n)$ where i is the index of a token selected for masking, x_i may be replaced with [MASK], a random token, or left unchanged according to the masking strategy described above.

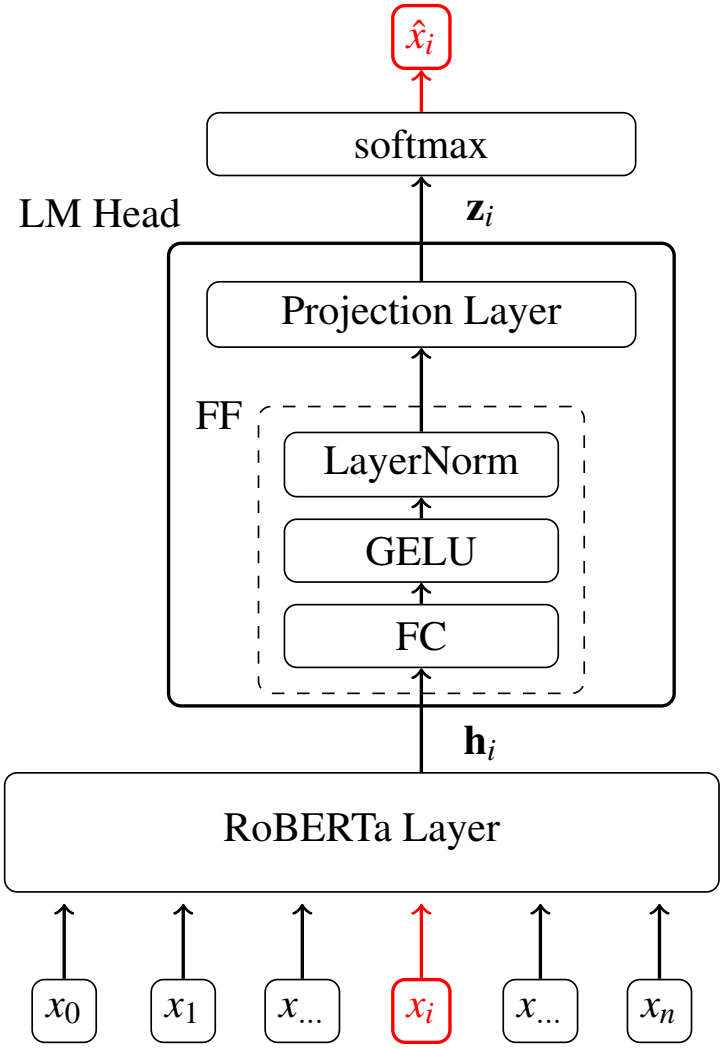

Figure 2.2: Overview of masked token prediction in RoBERTa.

RoBERTa contextualizes the masked token as follows:

$$\mathbf{H} = \text{RoBERTa}(\mathbf{x}), \tag{2.4.1a}$$

$$\mathbf{h}_i = \text{Extract}(\mathbf{H}, i), \tag{2.4.1b}$$

where $\mathbf{H}$ denotes the sequence of hidden representations produced by the multi-layered bidirectional transformers[2] of RoBERTa, depicted as RoBERTa Layer in the Figure 2.2, and $\mathbf{h}_i$ is the hidden representation corresponding to the i-th token. Extract in Equa-

[2] 12 layers for RoBERTa-base and 24 layers for RoBERTa-large.

tion (2.4.1b) is a function that selects the hidden state at position i from $\mathbf{H}$.

Next, the hidden representation of the masked token, $\mathbf{h}_\text{i}$, is fed into an LM head as in the figure to generate a deeper representation for the final masked token prediction. Formally,

$$\mathbf{z}_i = \text{LM Head}\big(\mathbf{h}_i\big), \tag{2.4.2a}$$

$$\text{LM Head}\big(\mathbf{h}\big) = \mathbf{W}_v\ \text{FF}(\mathbf{h})\ +\ \mathbf{b}_v, \tag{2.4.2b}$$

$$\text{FF}(\mathbf{h}) = \text{LayerNorm}\left(\text{GELU}\big(\mathbf{W}_f\,\mathbf{h}\ +\ \mathbf{b}_f\big)\right), \tag{2.4.2c}$$

where $\text{FF}(\cdot)$ denotes a feedforward network that serves an intermediate representation computed by a fully connected layer (FC in the figure), parameterized by the weight matrix $\mathbf{W}_f$ and the bias term $\mathbf{b}_f$, followed by a GELU activation (Hendrycks and Gimpel, 2016) and layer normalization (Ba et al., 2016). The weight matrix $\mathbf{W}_v$ and the bias term $\mathbf{b}_v$, project this intermediate vector into the vocabulary size, depicted as Projection Layer in the figure, thus making it suitable for token prediction as follows:

$$p(\hat{x}_i|\mathbf{x}) = \text{softmax}\big(\mathbf{z}_i\big). \tag{2.4.3}$$

In Figure 2.2, we illustrate the stacked bidirectional transformer component, referred to as "RoBERTa Layer". We do so because $\mathbf{h}_i$ in Equation (2.4.1b), the hidden representation produced by this component, is widely used and commonly referred to as "RoBERTa embedding" (Kennington, 2021; Kaminska et al., 2021; Wang and Riddell, 2022; Kuznetsov et al., 2024). The same applies to other MLM architectures, such as BERT, which also employ an LM head for the masked language prediction task; the hidden representation from the bidirectional transformer is frequently used in related work, including studies on semantic change (Kulkarni et al., 2015; Hamilton et al., 2016; Giulianelli et al., 2020; Martinc et al., 2020; Nagata et al., 2023) and metaphor detection (Choi et al., 2021; Li et al., 2023,).

However, as previously reviewed, the LM Head in Equation (2.4.2b) is the module the model actually predicts the masked token, and the $\text{FF}(\cdot)$ refines the context for the final masked token prediction. Despite this crucial role, its functionality is less widely recognized; only a few studies have explored the role of the LM Head in constructive decoding (Gera et al., 2023), accelerated inference (Arora et al., 2022; Elhoushi et al., 2024), and general masked token prediction (Petroni et al., 2019; Xu et al., 2020; Heinzerling and Inui, 2021; Gao et al., 2022). To effectively differentiate word usage, deeper repre-

sentations can better capture context, as supported by previous research on contextual embeddings (Ethayarajh, 2019). Therefore, we hypothesize that the feedforward output within the LM Head, which corresponds to the output in Equation (2.4.2c), provides a more context-rich representation than the commonly used "RoBERTa embedding", $\mathbf{h}_i$ in Equation (2.4.1b).

In this thesis, we leverage RoBERTa as a core methodology for detecting non-standard word usage. The capability of MLM to comprehend nuanced word usage within rich contextual environments makes RoBERTa particularly suitable for identifying deviations from standard patterns. Furthermore, we investigate the effectiveness of utilizing the LM Head in our experiments.

Chapter 3

Word Embedding-based Non-standard Word Usage Detection

In this chapter, we begin by investigating word usage through known non-standard word usage patterns found on the web. We then construct a dataset from X, a collection of user-generated social media texts, and manually annotate each instance to determine whether it is standard or non-standard usage. Finally, we propose a method that automatically detects non-standard usages by leveraging word embeddings.

3.1 Non-standard Word Usages in Japanese

3.1.1 Target Word Selection

In this study, we aim to investigate unconventional word usage. To achieve this, we first identify a set of candidate words, hereafter referred to as **target words**, for which we will annotate each occurrence in a sentence as either standard or non-standard. We selected target words that meet all of the following conditions:

1. The word has at least one known non-standard usage in the domains of computers, company/service names, or internet slang.
2. There exists an online resource or reference detailing as non-standard word usage.
3. The word appears at least 100 times in a balanced corpus (ensuring sufficient frequency).

Based on these criteria, we identified 40 total words: 10 from the computer domain, 10 from company/service names, and 20 from internet slang. Table 3.1 lists all 40 target words along with descriptions of their non-standard usages.

Domain	Word	Description of Non-standard Usage
Computers	串	Proxy
	鶴	Tool
	垢	Account
	蔵	Client
	尻	Serial Number
	鯖	Computer Server
	洒落	Share (software)
	狐	Firefox (software)
	泥	Android (operating system)
	窓	Windows (operating system)
Companies/Services	渋	pixiv
	支部	pixiv
	庭	KDDI
	林檎	Apple Inc.
	禿	SoftBank
	茸	NTT DOCOMO
	芋	eMobile (now known as Y!mobile)
	蟹	KLab Inc.
	密林	Amazon.com, Inc.
	尼	Amazon.com, Inc.
Internet Slang	草	Means “(laugh).”
	藁	Means “(laugh).”
	虹	Means “2D (two-dimensional).”
	惨事	Means “3D (three-dimensional).”
	乙	Means “good job” or “thanks for your work.”
	裏山	Means “jealous.”
	円盤	Means “DVD or Blu-ray disc.”
	凸	Means “attack/raid (online).”
	空気	Means “lacking presence.”
	駅弁	Refers to a regional national university.
	鉄板	Means “guaranteed” or “no doubt.”
	沼	Means “being stuck in a fandom/interest.”
	板	Means “online bulletin board.”
	安価	Means “reply,” derived from “anchor” rendered in Kanji.
	升	Means “cheat” (the Kanji 升 visually approximates “cheat”)
	ピザ	Refers to an overweight person or gaining weight.
	ゆとり	Refers to the “yutori” generation in Japan.
	地雷	Means “a person or thing that causes trouble.”
	養分	Invests time or money into another’s service.
	囲い	Pursues an online personality for potential offline contact.

Table 3.1: List of 40 selected words and their non-standard usage descriptions.

3.1.2 Observations

From Table 3.1, we identified several broad categories of non-standard word usages that characterize how Japanese internet users adapt or coin terminology.

In particular, many words can be grouped as follows:

- **Homophones**: The use of words that sound similar to their intended meaning is indeed common in Japanese internet slang. For example, "鯖 (saba, literally mackerel)" is used to represent "server" due to similar pronunciation.
- **Metaphorical Expressions**: The examples of "密林 (literally dense forest)" for the company Amazon and "円盤 (literally disc)" for DVDs/Blu-rays show how metaphorical associations are used. We also observed that some of the explanation of terms like "養分" and "囲い" accurately reflect their usage in specific online subcultures.
- **Visual Puns**: The observation about "升" representing "チート (cheat)" due to visual similarity when compressed and demonstrates the creative use of kanji in online communication.

Beyond these stylistic or structural patterns, an important functional division emerges:

1. **Referring to Existing Knowledge**: This category includes non-standard usages that anchor to recognized real-world entities or concepts but employ creative or alternate wording. For example, "林檎 (literally apple)" denotes Apple Inc., and "蟹 (literally crab)" refers to KLab Inc. Despite their unconventional linguistic forms, these expressions rely on widely understood referents (e.g., globally recognized companies or technologies).
2. **Referring to Non-existing or New Knowledge.** This category features expressions describing phenomena unique to online subcultures, lacking direct real-world analogues or dictionary-defined senses. A typical instance is "囲い (enclosure)", indicating being "an enthiams for physically meet (often within an online streaming community)". Such terms often originate within specific communities and may be unfamiliar to those outside these groups.

This creative range of non-standard usages in Japanese social media stems from phonetic similarities, metaphorical expansions, and playful orthographic alterations, all of which require careful consideration of context and community norms for accurate interpretation. Beyond highlighting linguistic ingenuity, these expressions also offer valuable clues for deeper semantic analysis, revealing insights into how meanings evolve, become disambiguated, and adapt to new or subcultural contexts. In this dissertation,

we focus primarily on the detection of non-standard word usage as our main task and therefore do not undertake semantic analysis or other closely related inquiries (Gella et al., 2013; Shoemark et al., 2017; Stewart and Eisenstein, 2018; Shoemark et al., 2019); nonetheless, we regard these findings as crucial for the future advancement of NLP. For example, learning to interpret phrase-describing models (Ni and Wang, 2017; Ishiwatari et al., 2019) or employing definition models (Noraset et al., 2017; Huang et al., 2021, 2022; Segonne and Mickus, 2023) can provide deeper insight into the actual meaning of expressions.

3.1.3 Data Collection and Human Annotation

To construct the dataset, we collected posts on X from January 1, 2016, to January 31, 2016, as our data source. Twitter was chosen because words often appear in both general and non-standard usages within the same platform, allowing us to capture diverse language phenomena. We performed morphological analysis on posts containing any of the 40 selected words and extracted 100 random posts for each word where it was tagged as a general noun. We employed MeCab[1] with the IPA dictionary[2] for morphological analysis. For each selected post, two annotators manually judged the usage of the target word in one of three ways: either in a literal sense, as part of a proper noun, or in a non-standard sense. If the target word was part of a proper noun (e.g., "井ノ尻," containing "尻"), the post was discarded from our final dataset. Additionally, if either annotator deemed the usage ambiguous or impossible to determine from context (96 posts in total)[3], it was excluded from the dataset.

Table 3.2 summarizes the resulting dataset with the distribution of standard versus non-standard usages across the three domains. The final dataset consists only of posts on which both annotators agreed, split into those judged *standard* and *non-standard* usages.[4] Cohen's kappa for this annotation task was 0.808, indicating substantial agreement. Because the dataset contains relatively few training examples for each word, it may not be suitable as a training corpus for supervised learning from the standpoint of data volume as we indicated in Chapter 1. Therefore, in this study, we propose an

[1] http://taku910.github.io/mecab/

[2] http://ipadic.osdn.jp/

[3] Examples include emoticon-like usage, e.g. "(´ 茸 `)," where the character "茸" ("mushroom") was simply part of a face emoji.

[4] Proper nouns were excluded because future named entity recognition methods could automatically remove such cases.

Category	#Standard	#Non-standard
Computers	416	234
Companies/Services	440	252
Internet Slang	817	814
Total	1,673	1,300

Table 3.2: The details of the labels for the created dataset on Japanese X.

Category	Words				
Non-standard Label Dominant	囲い	円盤	乙	垢	鯖
	支部	沼	草	裏山	密林
	地雷	凸	尼		
Standard Label Dominant	ピザ	安価	芋	駅弁	庭
	空気	串	狐	惨事	板
	尻	窓	蔵	林檎	洒落
	茸	鶴	渋	蟹	禿
Unbiased	藁	泥	鉄板	虹	養分
	升	ゆとり			

Table 3.3: Breakdown of the 40 annotated words.

unsupervised method to detect non-standard word usages and use this dataset solely for evaluation purposes.

We found that some words have a dominant label: one may be predominantly used in a standard manner, while another is largely used in a non-standard manner. Although each of the 40 target words was chosen for its known non-standard meaning, they do not all appear in non-standard usages at comparable rates. Table 3.3 categorizes the words into three groups based on their final labels for each usage:

- **Non-standard Label Dominant:** Over 70% of annotated posts were judged non-standard.
- **Standard Label Dominant**: Over 70% were deemed standard.
- **Unbiased**: No single label constituted more than 70% of annotations.

This grouping helps illustrate that some words, such as “鯖” for “computer server”, “草” to mean “laugh” are frequently used in unconventional senses, whereas others, such as “串” for “proxy” or “庭” for “KDDI”, more commonly retain their literal or dictionary-

Category	#Standard	#Non-Standard	Total
Non-standard Label Dominant	127	956	1,083
Standard Label Dominant	1,291	95	1,386
Unbiased	255	249	504
Total	1,673	1,300	2,973

Table 3.4: Distribution of standard and non-standard usages by label dominance.

defined meaning. Table 3.4 further quantifies how many standard vs. non-standard examples fall into each group.

3.2 Methodology

In this study, we propose a method for detecting non-standard word usage by considering the learning mechanism of SGNS (Skip-gram with Negative Sampling). Specifically, we begin by training Skip-gram with Negative Sampling (SGNS) on a balanced corpus to obtain distributed word representations. Next, we compute the inner product between the target word's vector and those of its surrounding words. If the computed value is low, the target word is judged to be used in a non-standard way. We employ a balanced corpus, a corpus consisting of texts sampled without bias to represent the language as a whole, to ensure that the learned word vectors capture the generality of word usage. Consequently, cases with high inner product values are interpreted as standard usages, while those with low values are considered non-standard usages.

Figure 3.1 provides an overview of the embedding-based method. In the proposed method, the word vectors are trained using SGNS. The non-standard usage of a target word is detected based on the weighted average similarity between the input-side word vector $\mathbf{v}^{IN}$ of the target word and the output-side word vectors $\mathbf{v}^{OUT}$ of its surrounding words. The weighted average of these similarity scores is defined as the word's standardness score. The reason for adopting the weighted average similarity as the standardness score is rooted in the SGNS learning process. SGNS applies weighted learning, giving greater emphasis to words closer to the target word in the context Levy et al. (2015).

We calculate the weight α for the weighted average as an integer using $\alpha = m + 1 - d$, where m is the window size and d is the distance from the target word to a surrounding word. Let the target word in the sentence be denoted by w_t, its input-side word vector by

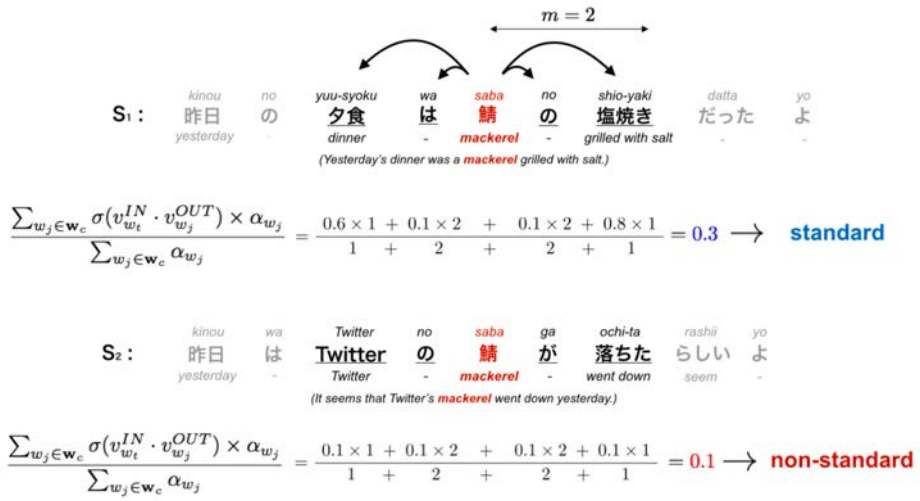

Figure 3.1: Overview of the word embedding-based method.

$\mathbf{v}^{IN}_{w_t}$, and the set of surrounding words within a window of size m by $\mathbf{w_c}$. The output-side word vector of each surrounding word $w_j \in \mathbf{w_c}$ is denoted by $\mathbf{v}^{OUT}_{w_j}$, and its associated weight is α_{w_j}. Given these definitions, the standardness score for the target word w_t is calculated as follows:

$$\frac{\sum_{w_j \in \mathbf{w}_c} \sigma(\mathbf{v}^{IN}_{w_t} \cdot \mathbf{v}^{OUT}_{w_j}) \times \alpha_{w_j}}{\sum_{w_j \in \mathbf{w}_c} \alpha_{w_j}}, \tag{3.2.1}$$

where σ is the sigmoid function. For unknown words appearing within the window, the word vector $\mathbf{v}^{OUT}_{unk}$ corresponding to the token for unknown words is used. This vector $\mathbf{v}_{unk}$ is derived from low-frequency words in the corpus used for training the word vectors.

The sigmoid function is used in Equation (3.2.1) because it is also employed as a non-linear function in the SGNS learning process[5]. If the computed standardness score is low, the target word's usage is judged to be non-standard, whereas a high score indicates standard usage.

[5]https://code.google.com/archive/p/word2vec/

3.3 Experiments

3.3.1 Baselines

The proposed method has the following three key features:

1. Use of a balanced corpus for learning word vectors.
2. Incorporation of both $\mathbf{v}^{IN}$ and $\mathbf{v}^{OUT}$ when computing the standardness score.
3. Application of a weighted average to refine the final score.

In the following sections, we introduce several baseline models, each trained under different conditions or with distinct representation learning strategies. This comparison enables us to assess how various design choices (such as the choice of training corpus or word-vector derivation method) impact the detection of non-standard usages. Through comparative experiments, we demonstrate how effectively these features enhance the detection of non-standard usages.

Models trained by different corpora: The proposed method calculates differences between the usages found in the training corpus and those found in the evaluation dataset, regarding these differences as a "standardness score." Since the word usages contained in the training corpus serve as a reference for detecting non-standard usage, we anticipate that the choice of training corpus will substantially affect detection accuracy. Accordingly, in this study, we prepare four corpora with different characteristics to determine which corpus best supports word-vector learning for this task. Table 3.5 presents an overview of each corpus.[6]

Models that only use $\mathbf{v}^{IN}$ by different representation learning algorithms: In contrast to common approaches that rely solely on $\mathbf{v}^{IN}$, our proposed method also utilizes $\mathbf{v}^{OUT}$. To evaluate the effectiveness of $\mathbf{v}^{OUT}$, we compare our method with a baseline that uses only $\mathbf{v}^{IN}$, following previous work Neelakantan et al. (2014); Gharbieh et al. (2016). In this baseline, $\sigma(\mathbf{v}^{IN}_{w_t} \cdot \mathbf{v}^{OUT}_{w_j})$ in Equation (3.2.1) is replaced by the cosine similarity, $\frac{\mathbf{v}^{IN\top}_{w_t}\mathbf{v}^{IN}_{w_j}}{\|\mathbf{v}^{IN}_{w_t}\|\times\|\mathbf{v}^{IN}_{w_j}\|}$, to compute the standardness score. Furthermore, to compare results with

[6]To compile the Web corpus, we followed the method of Kawahara et al. (2006) Kawahara and Kurohashi (2006). For Wikipedia, we used the Japanese edition as of July 2016, downloaded from https://dumps.wikimedia.org/jawiki/. The newspapers include the Mainichi, Nikkei, and Yomiuri from 1994 to 2004.

Corpus	Distinct Word Types	Word Count
BCCWJ	131,913	110 million
Web	336,048	600 million
Wikipedia	1,081,154	890 million
Newspapers	1,204,914	1.5 billion

Table 3.5: Corpora used for training. **BCCWJ**: Balanced Corpus of Contemporary Written Japanese (Maekawa et al., 2010). **Web**: Randomly extracted sentences from the Web. **Wikipedia**: Japanese Wikipedia (July 2016). **Newspapers**: Articles from 1994–2004.

word vectors learned by methods other than SGNS, we also conduct experiments using word vectors derived from Positive Pointwise Mutual Information (PPMI) via Singular Value Decomposition (SVD) Levy et al. (2015); Hamilton et al. (2016). In that setting, $\mathbf{v}^{IN}_{w_t}$ and $\mathbf{v}^{OUT}_{w_j}$ in Equation (3.2.1) respectively denote the t-th and j-th components after SVD. As with the baseline method, we use cosine similarity to compute the standardness score.[7]

Models that do not apply any weighting to the surrounding words: As described in Section 3.2, α in Equation (3.2.1) represents the weight corresponding to the distance between the target word and its surrounding words. To assess the effectiveness of this weighting, we compare against the case in which no weighting is applied, i.e., setting $\alpha = 1$ in Equation (3.2.1).

Models that discards function words in computing the score: We also conduct experiments that exclude function words, such as particles, auxiliary verbs, and conjunctions, from the set of surrounding words $\mathbf{w_c}$ in Equation (3.2.1). As illustrated in Figure 3.1, the surrounding words w_j in real text can be function words (e.g., particles). However, the similarity between the target word vector and function-word vectors may not necessarily be beneficial for determining the target word usage. Therefore, we also evaluate a model that calculates Equation (3.2.1) using the subset $\mathbf{w_{c'}} \subseteq \mathbf{w_c}$ from which these function words have been removed. In Mikolov et al. (2013), Skip-gram training is conducted under the assumption that high-frequency words carry less information than low-frequency words. Specifically, it employs sub-sampling of high-frequency words:

[7] In both the baseline that relies solely on $\mathbf{v}^{IN}$ and the SVD-based approach, we also tried using the sigmoid function in place of cosine similarity, but performance was lower than with cosine similarity.

Parameter	Hidden size
Input word vector	150
LSTM hidden layer	300
MLP input size	600
MLP hidden layer	600
Context vector	300
Output word vector	300

Table 3.6: context2vec model parameters.

each word in the training corpus is probabilistically excluded based on how often it appears. Because many function words fall into the high-frequency category, this approach can be seen as essentially similar to excluding function words.

A model that incorporates neural language model: Finally, to investigate whether deeper neural network layers influence detection accuracy, we compare our approach with context2vec (Melamud et al., 2016), a method for vectorizing context using multiple layers. While our proposed model relies on a shallow, single-layer neural network, context2vec employs a multi-layer model based on Bi-directional LSTMs (Bi-LSTMs). Another key difference is that Skip-gram predicts a target word based on its surrounding words in a fixed window, whereas context2vec processes the input sentence from left-to-right and right-to-left up to the target word. In context2vec, the entire sentence's word embeddings feed into a Bi-LSTM, whose left and right context outputs are concatenated before passing through a two-layer multilayer perceptron (MLP). The final output layer of the MLP (a context vector) is then used to predict the target word. In our task, the sigmoid-transformed dot product between this context vector and the target word vector serves as the standardness score.

3.3.2 Experimental Settings

We set the dimensionality to 300 for training word vectors. In each corpus, words with fewer than five occurrences were replaced with `<unk>` prior to training. For learning word vectors via SGNS (Skip-gram with Negative Sampling), we used the Python library `gensim` (Řehůřek and Sojka, 2010), setting the number of negative samples to 10. To train word vectors using SVD, we adopted the implementation by Levy et al. (2015)[8],

[8]https://bitbucket.org/omerlevy/hyperwords

where Positive Pointwise Mutual Information (PPMI) is first computed and then reduced via singular value decomposition. We trained for a total of five epochs. Additionally, we conducted experiments using window sizes of 2, 5, and 10.

For context2vec, we set the negative sampling number to 10 and used Adam (Kingma and Ba, 2014) as the optimization algorithm, with a learning rate of 10^{-3}. Table 3.6 lists details such as the dimensions of hidden layers and word vectors. To improve training efficiency, we treated BCCWJ and Web as medium-scale corpora, and Wikipedia and Newspapers as large-scale corpora. For the medium-scale corpora, we used 100 mini-batches, trained for 10 epochs, and replaced words with fewer than five occurrences with `<unk>`. For the large-scale corpora, we used 500 mini-batches, trained for 5 epochs, and replaced words with fewer than ten occurrences with `<unk>`.

For evaluation, we sort the standardness scores computed for each instance in the test set in ascending order. We then use average precision, taking the lower-ranked items in these scores to be classified as non-standard usages. In addition to these experiments, we also replicate the condition from Section 3.2 in which function words are not used.

3.3.3 Results

Table 3.7 presents the experimental results. In this table, "weighted" and "uniform" respectively indicate whether weighting is applied to the target word's surrounding words. The dagger symbol (†) denotes that the results show a statistically significant difference (at the 5% level based on a permutation test) when compared with the SGNS IN-OUT weighted model trained on BCCWJ, which is our proposed approach. Bold indicates, for each window size (2, 5, 10), the highest average precision among the SGNS IN-OUT, SGNS IN-IN, and SVD models.

Comparison for Model Selection

According to Table 3.7, the highest average precision is achieved by context2vec trained on Wikipedia, with a value of 0.845. Among our proposed methods, the SGNS IN-OUT weighted model trained on BCCWJ with a window size of 5 achieves the highest average precision at 0.839. Deeper neural network layers appear to contribute to accuracy, as evidenced by context2vec's stable, high values ranging from 0.803 to 0.845. Nevertheless, the results also demonstrate that even a shallow model—such as SGNS IN-OUT—can achieve performance comparable to deeper models when the learning approach, the

Training Corpus	Window	SGNS IN-OUT		SGNS IN-IN		SVD		context2vec
		weighted	uniform	weighted	uniform	weighted	uniform	
BCCWJ	2	.821	**.832**	.626†	.648†	.614†	.623†	.803
	5	**.839**	.833†	.734†	.736†	.673†	.650†	
	10	**.822**	.812†	.748†	.737†	.648†	.643†	
Web	2	.772†	.777†	.690†	.694†	.625†	.627†	.810
	5	.793	.787	.746†	.739†	.655†	.649†	
	10	.796	.786	.775	.761	.567†	.565†	
Wikipedia	2	.794†	.801	.712†	.724†	.630†	.632†	.845
	5	.775†	.767†	.781	.774	.628†	.626†	
	10	.782	.772	.719†	.695†	.730†	.734†	
Newspapers	2	.749†	.757†	.690†	.690†	.604†	.609†	.818
	5	.799	.791	.743†	.735†	.718†	.706†	
	10	.798	.784	.768	.754†	.730†	.718†	

Table 3.7: Average precision for each model.

handling of learned word vectors, and the choice of training corpus are suitably arranged.

Next, we compare performance for each window size. Focusing on window sizes of 2, 5, and 10, the SGNS IN-OUT model trained on BCCWJ attains the highest average precision at all three window settings. This outcome suggests that using BCCWJ and SGNS IN-OUT tends to yield high average precision regardless of window size. Additionally, for a window size of 2, 11 out of 12 models exhibit decreased average precision when weighted treatment is applied. A possible explanation is that, with fewer surrounding words, the weighting mechanism may overly emphasize function words near the target, leading to diminished accuracy. This point may be related to the results in later experiments where function words are excluded.

Comparison for Function Words

Table 3.8 shows the average precision achieved by models in which function words are excluded during the calculation of the standardness score. The dagger symbol (†) indicates that, at the 5% significance level under a permutation test, a statistically significant difference was observed compared with the SGNS IN-OUT weighted model trained on BCCWJ, which is our proposed method. From Table 3.8, we see that when function words are not used, the proposed SGNS IN-OUT weighted model yields the highest average precision. Specifically, this configuration uses BCCWJ as the training corpus, sets the window size to 5, and achieves an average precision of 0.857.

Training Corpus	Window	SGNS IN-OUT		SGNS IN-IN		SVD		context2vec
		weighted	uniform	weighted	uniform	weighted	uniform	
BCCWJ	2	.808†	**.810**†	.696†	.696†	.631†	.630†	.773†
	5	**.857**	.848†	.790†	.777†	.674†	.655†	
	10	**.849**	.839	.811†	.796†	.693†	.685†	
Web	2	.756†	.757†	.722†	.718†	.630†	.621†	.396†
	5	.794†	.786†	.753†	.742†	.669†	.664†	
	10	.773†	.759†	.793†	.783†	.711†	.699†	
Wikipedia	2	.778†	.778†	.733†	.734†	.636†	.631†	.776†
	5	.789†	.779†	.772†	.756†	.653†	.645†	
	10	.799†	.791†	.781†	.760†	.709†	.714†	
Newspapers	2	.755†	.745†	.722†	.718†	.630†	.624†	.808
	5	.799†	.791†	.760†	.750†	.714†	.700†	
	10	.808	.798†	.798†	.794†	.599†	.593†	

Table 3.8: Average precision for models that do not use function words.

Model	Training Corpus	Window	Weighting
SGNS IN-OUT	BCCWJ	10	×
			✓
SGNS IN-IN	BCCWJ	2	×
			✓
		5	✓
		10	×
			✓
	Wikipedia	10	×
			✓
SVD	BCCWJ	10	×
			✓
	Web	10	✓

Table 3.9: Models for which a significant difference was observed between function-word and non-function-word usage. A check mark (✓) indicates a weighted-model, whereas a cross mark (×) indicates a no-weight model.

From Tables 3.7 and 3.8, excluding function words during the calculation of the standardness score improved average precision in 48 out of 76 models. Of these, 12 models showed a statistically significant difference between excluding function words and including them. Table 3.9 lists the models for which a significant difference was

observed. In contrast, 26 models saw decreased average precision when function words were excluded, and 2 models showed no change. Notably, the average precision of context2vec decreased in four of these models. Since context2vec embeds words and then processes them via a Bi-LSTM and a multilayer perceptron, excluding function words might weaken its effectiveness as a language model, resulting in poorer predictions.

The model that showed the greatest improvement in score was the SVD weighted model trained on Web data with a window size of 10, yielding a 0.144-point improvement. On the other hand, the largest decrease in score, 0.414 points, occurred with context2vec trained on Web data. These findings suggest that how function words are handled can significantly affect average precision.

Next, we compare performance based on each window size. Referring to Table 3.8, when comparing the results at window sizes of 2, 5, and 10, the SGNS IN-OUT model trained on BCCWJ achieves the highest average precision under each condition, consistent with the broad trends noted in Section 3.3.3. In the experiments using a window size of 2, only 3 of the 12 models showed decreased average precision under weighting, which indicates that the adverse effects of weighting observed in Section 3.3.3 are mitigated. This outcome implies that smaller window settings may be more susceptible to the negative influence of function words.

Comparison for Weighting

From Table 3.7, examining how average precision changes between models that apply weighting (weighted) and those that do not (uniform) reveals that, among the 36 models evaluated, 22 show improved average precision when weighting is applied. In contrast, 13 models see decreased precision, and 1 model shows no change. Under the condition where function words are excluded (Table 3.8), 30 out of 36 models benefit from weighting in terms of average precision, 4 exhibit decreased precision, and 2 show no change. To investigate the significance of weighting, we conducted statistical tests on these experimental results. Out of 72 pairs of models (weighted vs. uniform), 36 pairs showed statistically significant differences in average precision. Table 3.10 summarizes these findings. Although not all configurations yielded statistically significant differences, our results indicate that surrounding words closer to the target word serve as valuable cues for this task, suggesting that the proposed weighting approach effectively leverages such local context.

Training Corpus	Window	SGNS IN-OUT		SGNS IN-IN		SVD	
		w/ FW	w/o FW	w/ FW	w/o FW	w/ FW	w/o FW
BCCWJ	2	×‡	×	×‡	×	×	×
	5	✓	✓	×	✓	✓	✓
	10	✓	✓	✓	✓	×	×
Web	2	×‡	×	×	×	×	✓
	5	✓	✓	×	✓	×	×
	10	✓	✓	✓	✓	×	×
Wikipedia	2	×‡	×	×‡	×	×	×
	5	×	✓	×	✓	×	×
	10	✓	✓	✓	✓	×	×
Newspapers	2	×‡	×	×‡	×	×	✓
	5	✓	✓	×	✓	✓	✓
	10	✓	✓	✓	×	✓	×

Table 3.10: Statistical investigation of weighting. "w/ FW" refers to models that include function words, and "w/o FW" refers to models that exclude them. A check mark (✓) indicates that the weighted model's average precision is statistically higher than that of the non-weighted model, whereas a cross mark (×) indicates no statistically significant difference was found in that comparison. A double dagger (‡) means the non-weighted model's average precision is statistically higher than that of the weighted model.

Model	**Training Corpus**	**Window**	**Weighting**	**Function Words**	**Avg. Precision**
SGNS IN-OUT	BCCWJ	5	✓	×	0.857
SGNS IN-IN	BCCWJ	10	✓	×	0.811†
SVD	Wikipedia	10	×	✓	0.734†
context2vec	Wikipedia	N/A	N/A	✓	0.845

Table 3.11: Experimental settings yielding the highest average precision for each model. A checkmark (✓) indicates that the feature is used, whereas a cross mark (×) indicates that it is not used. A dagger symbol (†) indicates that a statistically significant difference was found, at the 5% level via a permutation test, compared with the SGNS IN-OUT model.

Comparison of Highest Performance by Each Model

Table 3.11 summarizes the highest average precision scores attained by each model across all experiments, along with the corresponding experimental settings. Statistically significant differences emerged between the average precision of our proposed SGNS

IN-OUT model and both the SVD and SGNS IN-IN models. These results suggest that using $\mathbf{v}^{OUT}$ as well as $\mathbf{v}^{IN}$ helps improve average precision for this task.

We also performed a significance test comparing the SGNS IN-IN and SVD models; the resulting p-value was 0.089. Although SGNS-based methods appear to achieve higher average precision, the difference is not statistically significant.

Focusing on experimental configurations for each method, we see that SGNS-based methods tend to achieve higher average precision by using BCCWJ as the training corpus, applying weighting, and excluding function words. Conversely, methods based on SVD and context2vec achieve their best average precision by using Wikipedia as the training corpus and including function words. Because the most effective corpus for achieving higher average precision varies according to the particular learning method, choosing a corpus suited to each method is essential.

3.4 Oracle Performance under Real-world Settings

In our previous experiments, we evaluated document rankings within the evaluation dataset by sorting them according to their standardness scores, where higher scores indicate standard usage and lower scores indicate non-standard usage.

When we measure performance using average precision, we integrate performance across all possible score thresholds. Although ranking-based evaluations are valuable, real-world applications generally require selecting a single threshold to identify which usage is non-standard. Hence, we define a specific threshold for the standardness score, which separates standard from non-standard usage in a binary classification. In our setting, any instance whose standardness score falls below this threshold is labeled as positive (non-standard), while those that exceed the threshold are labeled as negative (standard). Because our primary interest lies in non-standard usage, we treat lower-scoring instances as "positive" (or "detected").

3.4.1 Word Usage Classification

In this section, to investigate upper-bound performance of each method in a real-world settings, we conducted an experiment by setting a specific threshold, and conduct error analysis for further improvements of the method. We classified instances with scores below this threshold as non-standard usage and instances with scores above it as standard usage, and evaluate the performance based on precision, recall, and F-values for non-

Model	Precision	Recall	F-score
SGNS IN-OUT	**.765**	.831	**.796**
SGNS IN-IN	.711	.814	.759
SVD	.603†	.805	.689†
context2vec	.743	**.852**	.794

Table 3.12: Results for the word usage classification task using the models that yielded the highest average precision, as listed in Table 3.11.

Model	Non-standard Label Dominant			Standard Label Dominant			Unbiased		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
SGNS IN-OUT	**.952**	.845	.895	**.237**	**.789**	**.364**	**.801**	**.791**	**.796**
SGNS IN-IN	.947	.834	.887	.192	.768	.307	.704†	.755	.729
SVD	.910	.840	.874	.126†	.758	.217†	.604†	.691	.644†
context2vec	.941	**.881**	**.910**	.202	.716	.316	.758	**.791**	.774

Table 3.13: Precision, Recall, and F-score for each class

standard labels. In this experiment, we tested thresholds ranging from 0.001 to 1.000 in increments of 0.001 and ultimately selected the threshold that produced the highest F-value for classification.

Table 3.12 shows classification results of each model. Bold values indicate the highest performance for each metric, and the dagger (†) indicates that the corresponding result is statistically significantly different from the proposed SGNS IN-OUT model at the 5% significance level, as determined by the randomization test. According to these results, the F-value obtained by the proposed method was the highest among all models at 0.796. Furthermore, the model that achieved the highest precision was SGNS IN-OUT (the proposed method), whereas the model that achieved the highest recall was context2vec.

Next, we investigated the evaluation metrics for each class shown in Table 3.3. In the constructed dataset, each word exhibits a degree of label imbalance, and Table 3.3 classifies 40 target words into three groups depending on whether they skew toward a particular label or show no skew. To analyze how the evaluation metrics vary by class, we computed the precision, recall, and F-value for each class. We used the same threshold as in the experiment described above.

Table 3.13 shows the results for each class. The dagger (†) again indicates that a statistically significant difference at the 5% level was confirmed by the permutation test,

when compared with the SGNS IN-OUT model. We observed that the precision in the non-standard label dominant class tends to be relatively high for all models, whereas precision for the standard label–dominant class tends to be low, indicating an imbalance in the evaluation metrics. These results suggest that choosing a threshold that maximizes the overall F-value succeeds in detecting words widely viewed as non-standard. However, words that only a small number of individuals consider non-standard often produce more false detections.

Next, we perform a qualitative evaluation of the experimental results produced by the proposed method. Below are examples of non-general word usages detected by the proposed approach:

(i) うちの場合**林檎**は父が触ったことないし、Android 端末いっぱい買って...
(ii) ... 国立**駅弁**よりすこし高いくらいかなでもそこにいったって...
(iii) 光村雨チケ今回何枚取れるかなー久しぶりだから**泥**率上げてくれるよね…？
(iv) ... やっぱり LT で他全員でガン**芋**してるのが一番強いんじゃないかな
(v) **鯖**落ちだあああああああああああああああああガチマに潜るなあああああああああ

In (i) and (ii), “林檎” and “駅弁” are examples where the proposed method successfully detected non-general usages, even for words whose general label is otherwise dominant. Moreover, in (iii) and (iv), “泥” and “芋” represent usages that differ from those shown in Table 3.2, yet the proposed method accurately detected them[9].

Example (v) was not detected by the method based on context2vec, yet it was successfully detected by the proposed method. Although context2vec considers the entire input sentence, in this particular instance the consideration of the full sentence appears to have led to an incorrect judgment. In contrast, because the proposed method focuses on the window of surrounding words before and after the target word, it is less affected by unrelated words in the sentence, leading to a successful detection.

3.4.2 Error Analysis

In this section, we present examples of non-standard word usages that were not detected by the proposed method in the usage classification:

[9] In this context, “泥” primarily refers to “drops” in social games, and “芋” primarily refers to “snipers” in online games.

Actual\Predict	Non-Standard	Standard
Non-standard	1080	220
Standard	332	1341

Table 3.14: Confusion matrix for the proposed method.

Actual\Predict	Non-standard	Standard
Non-standard	1108	192
Standard	384	1289

Table 3.15: Confusion matrix for context2vec.

(vi) ニコ動で実況者がワードバスケットやってて**草**

(vii) 55 連でテレーゼ、エクセ、ユイ、**虹**星 1。引きは微妙だけど一番欲しかった...

(viii) あ〜、なんだこの気持ち。変なの**藁**藁。醜い感情は押し殺せばいいか

(ix) 零十サンの規制してしまった時用**垢**。本垢フォローもよろしくでっす !!

(x) **養分**辞めたい吸収される側から…する側になるためには…カネが…カネが必要…！

As shown in (vi), there were cases in which the target word had very few surrounding words, and thus detection was unsuccessful. Nonetheless, as seen in (v), even when there are few surrounding words, detection may still be successful. Hence, in situations where surrounding context is limited, the model's output appears to be unstable.

Next, in (vii), when there are low-frequency words or unknown words (e.g., "ユイ" or "エクセ") near the target word, detection can fail. The proposed method uses a pre-trained vector for unknown words whenever they appear. However, to handle cases like these, additional processing tailored to unknown words may be necessary.

Furthermore, in (viii) and (ix), a tendency was observed for detection to fail when the target word itself appears among its own surrounding words. In SGNS training, the dot product between a word's $\mathbf{v}^{IN}$ and $\mathbf{v}^{OUT}$ vectors tends to be relatively high[10], causing the overall standardness score to rise and eventually leading to a detection failure.

Finally, (x) illustrates a case that was not detected by the proposed method but was successfully detected by the context2vec-based method. Because the proposed method considers only the fixed window of surrounding words around the target word, it cannot account for clues, such as "カネ," which appears at a distance from the target word but may be essential for interpreting its usage. In contrast, context2vec considers the entire sentence, enabling successful detection.

[10]When calculating the dot products of $\mathbf{v}^{IN}$ and $\mathbf{v}^{OUT}$ for 10,000 randomly sampled words, the dot product for a given word's own $\mathbf{v}^{IN}$ and $\mathbf{v}^{OUT}$ was, in 9,997 out of 10,000 cases, higher than the average dot product between $\mathbf{v}^{IN}$ and $\mathbf{v}^{OUT}$ for other words. Note also that the cosine similarity between $\mathbf{v}^{IN}$ vectors for the same word is always 1, so the SGNS IN-IN model experiences a similar issue.

We further perform an error trend analysis using confusion matrices for the experimental results presented in Table 3.12. In addition to the proposed method, we also analyze context2vec, which demonstrated high performance in previous experiments in Tables 3.11 and 3.12. Tables 3.14 and 3.15 show the confusion matrices corresponding to each set of experimental results. The top-right cell in each confusion matrix indicates false-positive instances, where the model incorrectly predicted a standard usage as non-standard. Conversely, the bottom-left cell represents false-negative instances, where the model failed to detect a non-standard usage and instead labeled it as standard. From Table 3.14, the proposed method produced 220 false negatives (missed detections) and 332 false positives. Meanwhile, Table 3.15 shows that the context2vec-based approach yielded 192 false negatives and 384 false positives. Comparing these results, the proposed method has relatively more false negatives and fewer false positives, whereas the context2vec-based method exhibits the opposite pattern.

In summary, the proposed method exhibits fewer false positives but more false negatives compared to the context2vec-based approach. From a practical standpoint, the choice between these two methods depends on whether it is more critical to minimize missed detections of non-standard usages (favoring context2vec) or to avoid falsely flagging standard usages as non-standard (favoring the proposed method). Which type of error introduces more risk depends on the specific application. For instance, in constructing a dictionary of non-standard usages, overlooking (false negatives) can be more problematic, because missed usages are difficult to recover later. On the other hand, in a live dialog system, erroneous interpretations of standard usages as non-standard (false positives) may undermine user trust more than failing to detect non-standard usages. Therefore, depending on the intended use, one approach may be more suitable than the other.

3.5 Discussion

We introduced an approach that uses local contextual cues and function-word exclusion to detect non-standard word usages effectively. Here, we focus on two substantial considerations in our design that may restrict its versatility in practice.

Global threshold: A single static threshold overlooks the fact that non-standard usages often manifest differently across distinct domains and user groups. The use of a global threshold may, for example, lead to over-detection of non-standard usages in more formal

corpora, while under-detecting them in more colloquial or domain-specific data. This phenomenon surfaced in our error analysis, particularly when comparing results on BCCWJ (a balanced corpus) versus Web data (with more slang). Although the global threshold achieved high precision in certain domains, we observed that specialized texts with unique jargon (or distinct usage patterns) sometimes slipped through undetected, resulting in higher false negatives.

Fixed window size: The proposed model incorporates a fixed window size for the surrounding context, which imposes artificial boundaries around each target word and potentially misses important context beyond that window. As illustrated in example (x) in the error analysis, the target word "養分" was not flagged as non-standard because the relevant clue, "カネ", appeared several tokens away, outside the fixed window. In contrast, a model such as context2vec, which considers the entire sentence via a Bi-LSTM, successfully detected that usage. While our weighting and function-word exclusion strategies partially mitigate these constraints (for instance, by emphasizing words nearer to the target), they do not fully resolve instances where critical signals lie far from the immediate neighborhood.

Together, these limitations underscore the need for a more adaptive approach that can flexibly adjust to domain variation, rather than relying on a single global threshold, such as in classification-based models built on a language model (Peng et al., 2003; Chen et al., 2022) to fully incorporate broader contexts.

3.6 Conclusion for this Chapter

In this study, we developed an automatic approach to detect non-standard word usages, focusing on social media terms whose meanings may not appear in existing dictionaries. To support our research, we introduced a manually annotated dataset of 40 Japanese words, each of which has at least one non-standard meaning. Every usage of these words in the dataset was then manually annotated as either standard or non-standard by domain experts. The proposed method employs Skip-gram with Negative Sampling to learn word vectors from a balanced corpus. We then compute a "standardness score" for each target word usage by taking a weighted average of the sigmoid-transformed dot products between the target word's vector and those of its surrounding words. If the standardness score is high, the usage is deemed standard; if it is low, the usage is deemed non-standard. In our method, rather than using only $\mathbf{v}^{IN}$, which is common in prior

research, we combined $\mathbf{v}^{IN}$ with $\mathbf{v}^{OUT}$ for this computation. Experimental results show that incorporating $\mathbf{v}^{OUT}$ and weighting surrounding words by proximity significantly improve detection accuracy, suggesting that nearby context offers vital cues. Further improvements were achieved by excluding function words, indicating that function words may not be crucial for detecting non-standard usages.

This research serves as a starting point for analyzing words used in non-standard ways on social media. By extending the proposed approach, we anticipate that our findings will facilitate broader analyses of word usage across different social media data. Although our evaluation experiments were conducted in a closed setting using the constructed dataset, applying threshold-determination strategy to the standardness scores should enable open settings, such as extracting non-standard usages from previously unseen data, which we address in the following chapter.

Chapter 4

Masked Language Model-based Non-standard Word Usage Detection

Chapter 3 introduced a Japanese dataset for non-standard word usage detection and proposed a word embedding-based method for this task. In this chapter, we present a masked language model-based approach that addresses the limitations of the word embedding-based method, namely the need for a threshold and limited access to context words, by employing a classifier-based strategy. We begin by revisiting the task definition for clarity.

4.1 Task Definition

This study addresses a **word usage classification** task, where we are given an input sentence and a target word within that sentence. The goal is to determine whether the usage of the target word is *standard* or *non-standard*. We then extend this approach by applying the same classification scheme to all target words in the input sentence, thereby transforming a single-word classification task into a **word usage detection** task that identifies any non-standard usages across multiple words. In this study, we lay the groundwork for our approach by focusing exclusively on *nouns*, leveraging their intrinsic semantic importance as an initial lens through which to investigate word usage classification.

Figure 4.1 illustrates an example word usage classification task on a given sentence and corresponding target word. Specifically, the model inspects the contextual usage of the target word and decides whether that usage aligns with a dictionary definition (*standard*) or diverges from it (*non-standard*). In the provided example, the word *nova* is used

Model Input
Sentence: *If you were actually good enough to be in low* ***nova****, then you wouldn't be in that level.* **Target Word**: *nova*
Model Output: *non-standard*

Figure 4.1: Example input and output for the word usage classification task.

as *non-standard* because it refers to a specific gaming rank rather than its conventional astronomical meaning.

4.2 Methodology

We assume that for non-standard usages, both (1) predicting the target word from its surrounding context and (2) predicting the context words from the target word present challenges. However, we hypothesize that these prediction tasks provide important clues for detecting non-standard usages. Based on this insight, we propose a model that incorporates vector representations produced from both *context2target* prediction task, where we predict a target word from its surrounding context words, and *target2context* prediction task, where we predict the surrounding context words from the target word. By combining the information from these two approaches, our model seeks to enhance the accuracy of identifying non-standard word usages. In this section, we briefly review methods for *context2target* prediction and *target2context* prediction commonly used in NLP.

context2target prediction

Masked language models such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019) are trained to predict masked tokens from a given context.[1] A masked language model takes as input a masked sentence such as $X' = \{w_1, w_2, [MASK], w_4, .., w_n\}$ (where the original token w_3 is masked as $[MASK]$), and predicts the ground-truth masked token w_3 from a contextual representation. The masked language model uti-

[1]CBOW (Mikolov et al., 2013) is an earlier well-known method for learning word embeddings based on *context2target* prediction. We include experiments based on CBOW in this paper, but find that masked language model performs better than CBOW.

lizes bi-directional transformers for capturing contextual representation, which are composed of multi-headed self-attention mechanisms and element-wise feedforward networks (Vaswani et al., 2017). As illustrated in Figure 2.2, on top of these transformer layers, both BERT and RoBERTa build a prediction layer known as language modeling head (LM head)[2], which is an architecture designed to predict masked tokens through a vocabulary-sized projection layer. This LM head is trained using the cross-entropy loss function, where the model learns to predict masked tokens during training. Combined with the softmax function, this layer generates a probability distribution over all possible tokens, allowing the model to predict the masked token by selecting the one with the highest probability. The LM head is implemented by first applying a fully-connected layer to the input features coming from bi-directional transformers, followed by the GELU activation function (Hendrycks and Gimpel, 2016) and layer normalization (Ba et al., 2016). After that, the processed features are projected back to the size of the vocabulary through a projection layer that includes a bias term. This architecture plays a crucial role as a masked language model, enabling the prediction of masked tokens from the given context.

As mentioned in Chapter 2, since the masked language model directly models target word prediction based on context, it can be used to capture difference of training contexts for a given target word. Thus, in our model, we use contextualized representations obtained from a masked language model as a feature for *context2target* prediction. In the experiment, we used the hidden representation obtained from the LM head by excluding the final projection layer. Specifically we utilize the hidden representation obtained from Equation (2.4.2c), the output after applying the fully-connected linear layer, GELU activation, and layer normalization, and finetune for the classification task.

target2context **prediction**

As described in Chapter 2 and 3, Skip-gram Negative Sampling (SGNS) (Mikolov et al., 2013) is a method for learning word embeddings based on predicting context words from a target word in vector space, by assigning two different vector representations to every word in the vocabulary: one for that word as a target word, and one as a context word. Given a target word in context, SGNS is trained by maximizing the dot-product between the vector for the target word and vectors for the context words.

[2]We follow an implementation by HuggingFace. https://github.com/huggingface/transformers/blob/main/src/transformers/models/roberta/modeling_roberta.py#L1116

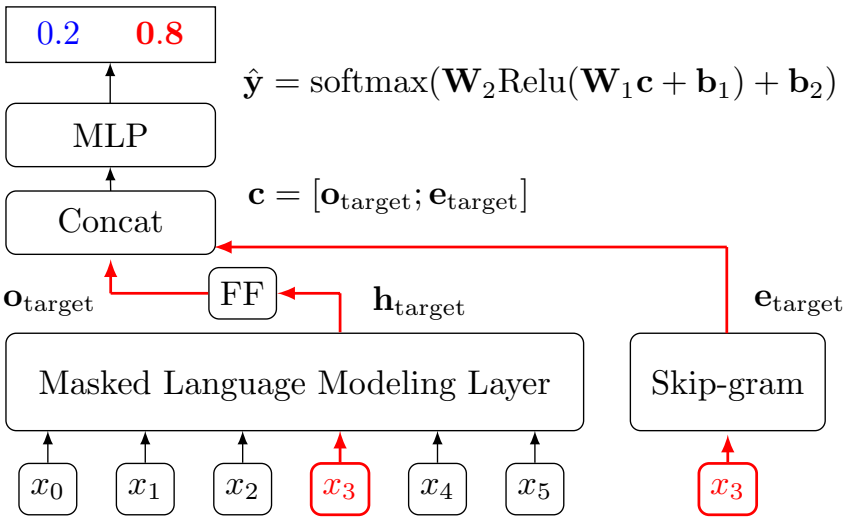

Figure 4.2: Overview of the MLM-based method.

We regard this training procedure as *target2context* prediction, where we assume that the vectors trained by *target2context* prediction will produce a different aspect to the vectors obtained from *context2target* prediction. As proposed in Chapter 3, this training procedure can also model how a target word fits in context by manipulating the trained vector representations, suggesting that *target2context* prediction is helpful in classifying word usages. Based on this finding that those two different representations from SGNS called *IN* and *OUT* embeddings play an important role in distinguishing word usages, we also incorporate these two distinct vectors into our model.

4.2.1 Word Usage Classifier

In this work, we classify a target word in a sentence as either standard or non-standard using a binary word classifier which takes into account of contextual features from *target2context* prediction and *context2target* prediction. For the task formulation, given a sentence with n words $X = \{w_1, .., w_n\}$ and a target word $w_{\text{target}} \in X$, we classify w_{target} as $y \in \{0, 1\}$, where 0 indicates a standard usage and 1 indicates a non-standard usage.

Figure 4.2 presents an overview of our proposed method. We first feed the sentence X into the masked language modeling layer to get an initial contextualized representation $\mathbf{h}_{\text{target}}$. We then feed this representation to the feedforward network FF($\cdot$), which is a

part of the component of the LM Head as in Equation (2.4.2c), to make a further contextualized representation. We treat this output $\mathbf{o}_{\text{target}}$ as a *context2target* representation of w_{target}.[3] We use the mean of the contextualized embeddings for each of the wordpiece tokens that make up the target word. Note that we keep the target word unmasked for the training of non-standard word detection, (i.e. we do not replace the target word w_{target} with $[MASK]$), as we find that masking the original word negatively impacts on results, consistent with the findings of Zhou et al. (2019) in the context of lexical substitution. We also incorporate a *target2context* representation $\mathbf{e}_{\text{target}}$, in the form of a word embedding of the target word w_{target} learned by Skip-gram . We concatenate the two vectors to form $\mathbf{c} = [\mathbf{o}_{\text{target}};\ \mathbf{e}_{\text{target}}]$, as the combined representation of the target word. Finally, we feed $\mathbf{c}$ into a 2-layered multi perceptron with ReLU activation (Agarap, 2018) and softmax function to implement binary classification, as follows:

$$p(\hat{\mathbf{y}}|X) = \text{softmax}(\mathbf{W}_2\,\text{ReLU}(\mathbf{W}_1\mathbf{c} + \mathbf{b}_1) + \mathbf{b}_2), \tag{4.2.1}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, $\mathbf{b}_2$ are weight matrices and bias vectors. To train the classifier, we use binary cross entropy loss:

$$L(y, \hat{y}) = -\frac{1}{D}\sum_{i=0}^{D} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i), \tag{4.2.2}$$

where $y_i \in \{0, 1\}$ is the gold label, $\hat{y}_i$ is the model prediction (i.e. probability of non-standard usage), and D is the number of examples in the minibatch.

Similar fields incorporate features from masked language models such as BERT, but few utilize the feature from the LM head. Metaphor detection models also incorporate features from masked language models (Choi et al., 2021; Li et al., 2023,). In these studies, dedicated networks are proposed based on the Metaphor Identification Procedure Crisp et al. (2007) and Selectional Preference Violation (Wilks, 1975, 1978), using BERT embeddings derived from the output of a bidirectional transformer. Our proposed method, however, does not employ specialized networks for these tasks. Instead, we introduce features based on skip-gram and use the information from the LM head following the transformer output, rather than relying on the transformer output itself. Similarly, in the field of diachronic semantic change, many studies utilize contextualized embeddings from models like BERT. Their primary focus, though, is on using word

[3]Each word is tokenized into wordpieces to reduce the vocabulary size.

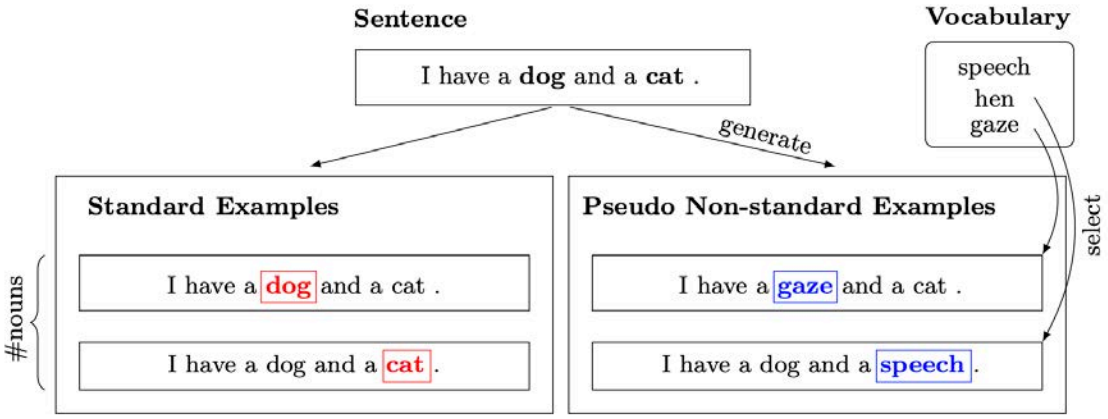

Figure 4.3: Pseudo-example creation.

similarity calculations across different time spans Kulkarni et al. (2015); Hamilton et al. (2016); Giulianelli et al. (2020); Martinc et al. (2020); Nagata et al. (2023) or clustering with topic models Montariol et al. (2021); Inoue et al. (2022). This focus differs from our classification-based approach.

4.2.2 Pseudo-label Training

Equation (4.2.2) requires the gold label y to train the model, meaning that we need a large-scale source of annotated data. However, constructing a large-scale dataset is expensive as it needs expert lexicographic knowledge. To counter the lack of manually-annotated training data for this task, we incorporate the idea of pseudo-example learning (Bergsma et al., 2008; Poon et al., 2009; Pauls and Klein, 2012; Kiyono et al., 2019), where we automatically generate a dataset of synthetic non-standard examples from a raw corpus, and use it to train the model. That means that our model does not need any annotated training data and can be trivially adapted to different domains and languages, given that we can easily generate training data given a monolingual corpus.

Figure 4.3 depicts the method for constructing the training dataset with pseudo examples. The procedure is composed of two steps: (i) choose a target word in a sentential context as a standard example; and (ii) replace the selected word with an alternative word based on uniform random sampling from vocabulary in the training corpus, and regard the modified sentence as a non-standard example. Note that we iterate the target word selection over all possible nouns in each sentence, meaning that the same sentence is repeatedly used during training based on the number of nouns it contains. Since this approach does not require any human annotation, we can train the model on any language, and present experiments on both English and Japanese.

4.3 Experiments on Japanese Social Media Dataset

4.3.1 Experimental Settings

We built a Japanese Twitter dataset in Chapter 3, where a target word in each sentence is manually annotated as to whether its usage in context makes sense relative to common-sense expectations. As reported in Chapter 3, some target words in the dataset are not used as nouns but rather as parts of named entities. Therefore, we use a subset of this dataset in our experiments, after filtering out any word segmentation errors. To perform the word segmentation filtering, we used two Japanese morphological analyzers — Mecab (Kudo et al., 2004) and Juman++ (Tolmachev et al., 2018) — and selected those target words which both analyzers identified as a single-morpheme common noun. The filtering process eliminated 385 sentences from the original dataset, resulting in a total of 2,588 sentences, consisting of 1,380 standard usages and 1,208 non-standard usages. Note that this X dataset is only used for evaluation and not for training, fine-tuning, or hyper-parameter tuning. To train our model, we took the BCCWJ corpus (Maekawa et al., 2010) and extracted one million (standard) examples from the corpus, from which we generated one million pseudo-examples according to the process described in Figure 4.3. For hyper-parameter tuning, we prepared a separate validation dataset from BCCWJ made up of 1000 sentences evenly split between standard and pseudo non-standard examples.

As evaluation metrics, we use accuracy and average precision to measure overall classification performance, and precision, recall, and F1 score to assess detection performance specifically for non-standard usages. We report averaged performances across three different random seeds.

4.3.2 Model Details

Following Chapter 3, we use word embeddings via Skip-gram with negative sampling over the BCCWJ corpus (Maekawa et al., 2010), with dimensionality 300, window size 5, and the number of negative samples set to 10, which resulted the best performance in the previous experiments. We also incorporate *IN* and *OUT* embeddings produced from Skip-gram forming *target2context* representation in Equation (4.2.1) as $\mathbf{e}_{\text{target}} = [\mathbf{v}^{IN}_{\text{target}}; \mathbf{v}^{OUT}_{\text{target}}]$ where $\mathbf{v}^{IN}_{\text{target}}$ and $\mathbf{v}^{OUT}_{\text{target}}$ are the *IN* and *OUT* embedding for the target word. In regard to this point, we provide an ablation study on feature representation for a target

word. For the masked language model, we use pre-trained Japanese RoBERTa$_{\text{base}}$ [4][5], which was trained over Japanese Wikipedia and the Japanese portion of CC-100. During training, we freeze the pre-trained weights of the word embeddings and bi-directional transformers in RoBERTa, and only finetune a part of LM head described in Section 4.2 and the word usage classification layer.

In the proposed method, we apply dropout (Srivastava et al., 2014) before the ReLU activation in Equation (4.2.1) with a dropout rate of 0.1. The hidden size for the affine transformation in Equation (4.2.1) is set to 768. We used the Adam optimizer (Kingma and Ba, 2014), with learning rate 0.001, learning rate decay factor 0.99, and minibatch size 128. We trained the model for 30 epochs, and performed model selection based on the model that achieved the best F1 score on the validation dataset. All experiments were done by using HuggingFace (Wolf et al., 2020).

4.3.3 Baselines

We prepared comparison methods for this word usage classification task. In this experiment, all models address the same problem: classifying a target word in the given sentence as either standard or non-standard. We evaluate the performance of the model relative to two baseline methods as detailed below.

1. WORD2VEC: This method was proposed in Chapter 3, and models the non-standardness of a target word in context according to a weighted average of the dot-product between the word embedding for the target word and word embeddings for the surrounding words (based on a fixed context window size). This method requires a global threshold to classify whether a given target word is used in a standard way or not. In the experiment, the threshold is determined by 10-fold cross validation over the data. In comparison, our model does not rely on any such threshold for classification as it is trained directly over the end-task of standard vs. non-standard usage prediction.

2. T5$_{\text{base}}$: To investigate the effectiveness of a pre-trained masked language model for this task, we employed Text-to-Text Transfer Transformer (T5) (Raffel et al., 2020), a large language model that does not incorporate a specific masked language modeling architecture within its design. We used a pre-trained Japanese T5$_{\text{base}}$[6],

[4]https://huggingface.co/nlp-waseda/roberta-base-japanese
[5]Due to computational resources and limitations, we use the base model instead of the large model.
[6]https://huggingface.co/sonoisa/t5-base-japanese

which was trained over Japanese Wikipedia and the Japanese portion of CC-100 and OSCAR. Note that we did not use T5's original decoder, relying solely on the encoder's hidden layers for this task. For a fair comparison, we added the same LM head networks used in RoBERTa (Section 4.2) to the encoder of T5, applied the same classifier layer afterward, and incorporated *IN* and *OUT* embeddings in the same manner as in the proposed model. As with the proposed model, we freeze the pre-trained weights of the T5 encoder and word embeddings, resulting in the same set of trainable parameters. The training corpora and parameters were likewise consistent with those used in the proposed model.

We also report the metrics produced by large language models (LLMs), such as ChatGPT (OpenAI, 2023). However, it is important to note that direct comparison with these models is not feasible due to the financial and computational costs associated with applying such methods to large-scale text streams, like those from Twitter or Reddit. These limitations result in a lack of scalability and make it challenging for these methods to rank examples based on their level of non-standardness. For the comparison with the LLM based method in our task, we employed GPT-4[7] (OpenAI, 2024) and Swallow-70b instruction model[8] (Fujii et al., 2024), both of which are multilingual models capable of handling Japanese texts. In these methods, we instructed the model to determine whether the target word in the input text was used in a standard or non-standard way. For constructing a prompt, we incorporated a few-shot approach (Bsharat et al., 2024; Tarumoto et al., 2024), providing examples that included both standard and non-standard usages to boost the performance. Table 4.1 shows the concrete prompt used in the experiment. Responses indicating agreement or disagreement were converted into binary labels for evaluations.

4.3.4 Results

Table 4.2 shows the results on the Japanese Twitter dataset. The proposed method achieves overall best performance, balancing high accuracy, F1 score, and average precision. The proposed model outperformed the second-best baseline, $T5_{base}$, by a margin of 1.2 to 2.5 points across metrics. Based on the results, our proposed method outperforms all baseline models in terms of accuracy, F1 score and average precision at a level of statistical significance ($p < .01$) by permutation test.

[7]Specifically, we used the `gpt-4-0613` model in batch inference.

[8]We used `tokyotech-llm/Swallow-70b-instruct-hf` through HuggingFace library.

Prompt

指示: 文中で与えられた単語が一般的な用法かどうかを答えてください。一般的な用法の場合「はい」を、そうでない場合は「いいえ」を答えてください。

入力:
文章: この仮面かわいいなあ〜
単語: 仮面
回答:
はい

入力:
文章: 仮面して東工大目指そうかな
単語: 仮面
回答:
いいえ

入力:
文章: 北海道産のバニラアイス甘くて美味しい
単語: アイス
回答:
はい

入力:
文章: 関東圏にて手押しで極上アイス売り始めました
単語: アイス
回答:
いいえ

入力:
文章: {*sentence*}
単語: {*target_word*}
回答:

Table 4.1: Japanese prompt used for GPT-4 and Swallow.

Compared to WORD2VEC, $T5_{base}$ and the proposed model achieves better results in major metrics, highlighting the importance of an explicit word usage classification layer in more parameterized pre-trained language models and pseudo-label training. Specifically, when evaluating accuracy in relation to the number of words in a sentence,

Model	Accuracy	Precision	Recall	F1	Average Precision
WORD2VEC (SGNS IN-OUT)	.752	.752	**.834**	.793	.860
$T5_{base}$	.816	.799	.808	.803	.877
Proposed Method	**.831**	**.811**	.833	**.821**	**.896**
Swallow-70b	.609	.555	.823	.663	-
GPT-4.0	.805	.830	.685	.750	-

Table 4.2: Experimental results on the Japanese Twitter dataset. **Bold** indicates the best score.

the top 20% of longer sentences, containing the most words, achieved accuracies of .810 and .830 in the WORD2VEC method and the proposed method respectively, while the bottom 20% of shorter sentences, containing the fewest words, had accuracies of .781 and .793, respectively. This indicates that the absolute difference in accuracy is greater in the top 20%. One possible explanation for this is that WORD2VEC method has a limitation due to its cut-off window, where the capacity of the model is restricted by the size of its context window, resulting in the discarding of valuable information in sentences that contain more words than the window can accommodate. In contrast, our proposed model fully leverages the contextualized representation from the bi-directional transformer, leading to higher accuracy for longer sentences. Shorter sentences were more difficult to classify for both methods due to the lack of contextual information, leading to lower performance. This aligns with the observations made by Pei et al. (2019) in slang detection. Compared to $T5_{base}$, the proposed model outperformed all the evaluation metrics suggesting that an explicit masked language modeling architecture helps in this task. GPT-4.0 and Swallow did not perform well in the classification. Further prompt engineering could help those models achieve better quality, but this is beyond the scope of this paper.

Table 4.3 shows a quality comparison of each method. In contextually rich examples, such as the first and second examples, all models were able to correctly predict the target words. These sentences provide strong contextual cues that make it easier for the models to infer the intended meaning of the target words. WORD2VEC model fails to predict correctly when the target word appears at the beginning or end positions in the sentence. This limitation is due to the cut-off window inherent in the Skip-gram architecture, which restricts the contextual information to a fixed window size around

WORD2VEC	$T5_{base}$	Proposed	GPT-4	*Sentences*
✓	✓	✓	✓	色々考えた結果、**鯖**移動しようかと。
✓	✓	✓	✓	スタイラスペン買おうと思うんだけど、**尼**のランキングだと聞いたことないメーカーのがいっぱいだな……
	✓	✓	✓	**支部**のアクセス解析見てみたら三割のかたが「鬼白ヤンデレ」で辿り着いてたけどどういうことですか？ ...
	✓	✓		名刺と**支部**には書いてるから知ってる方もいるかなと思うけども
		✓		複**窓**できるならそれに越したことないけどね
				完全に**沼**に沈みました
				毎回何故か音 MAD があって**草**生やしちゃうやばいやばい

Table 4.3: Example sentences and corresponding model predictions. **Bold** indicates the target word. A checkmark (✓) denotes a correct model prediction.

the target word. When the target word is near the sentence boundaries, the model lacks sufficient context on one side, leading to incorrect or ambiguous interpretations. $T5_{base}$ model and the proposed method demonstrate superior performance in handling abbreviations and domain-specific terms, particularly in cases involving "支部", which refers to the platform name "pixiv" in these examples. All models struggle with short sentences that lack sufficient context or that incorporate commonly used expressions from the internet. As Pei et al. (2019) reported, investigating word usages for short sentences is generally difficult. In the latter case, we hypothesize that this issue arises from both the training corpora and the pre-trained knowledge of LLMs. This problem, particularly in the context of evolving word senses, is closely related to diachronic semantic change (Kulkarni et al., 2015; Hamilton et al., 2016; Kutuzov et al., 2018), a research field that investigates how the meanings of words shift over time. A potential solution may involve incorporating historical corpora, such as COHA[9], to better capture these temporal variations in meaning (Hamilton et al., 2016; Sommerauer and Fokkens, 2019; Martinc et al., 2020; Nagata et al., 2023). However, such language resources are not available for the Japanese language. Alternatively, introducing a bias toward formal language by integrating formal written texts, such as newspaper articles, into the training

[9]https://www.english-corpora.org/coha

Model	Accuracy	Precision	Recall	F1	Average Precision
Proposed Method	**.831**	**.811**	**.833**	**.821**	**.896**
w/o Word Embeddings	.761	.754	.727	.740	.807
w/o Finetuning LM head	.810	.770	.844	.805	.875
w/o LM head	.820	.793	.831	.811	.885

Table 4.4: Ablation study on Japanese Twitter dataset. **Bold** indicates the best score.

data could help further mitigate this issue, which we propose as a direction for future research.

Table 4.4 shows the ablation study to understand the effectiveness of masked language model and word embeddings. All the ablation points demonstrated a statistical significance compared to the proposed model, with a level of $p < .01$ according to permutation test. As detailed in Table 4.4, utilizing both contextualized representations and word embeddings boosts classification performance, suggesting that it is helpful for the model to have features from both *context2target* prediction and *target2context* for this task. We also found that fine-tuning the LM head, which directly models a target word prediction (Section 4.2), was the second most crucial factor. This suggests that freezing the LM head restricts its potential ability to understand and distinguish given usages though pseudo-label classification. While the presence of the LM head itself shows the statistical margin, it was the least impactful compared to other ablation points. Overall, the ablation study indicates that both the masked language model and word embeddings significantly enhance task performance and these findings suggest that leveraging the full capabilities of the pre-trained prediction component, including the LM head, is important for better performance.

4.3.5 Error Analysis

We analyzed the error cases to investigate what misclassifications were made by the models. For the analysis, we picked the $T5_{base}$ model and the proposed model that performed the best F1 score on the testing set.

Table 4.5 shows confusion matrix for each model to analyze classification errors. In this study, we regard positive as non-standard usages and negative as standard usages. The top right cell represents the false-positive cases, where the model incorrectly predicted

Actual\Predict	**Standard**	**Non-standard**
Standard	1,046	334
Non-standard	195	1,013

a Confusion matrix for WORD2VEC (SGNS IN-OUT).

Actual\Predict	**Standard**	**Non-standard**
Standard	1,155	225
Non-standard	246	962

b Confusion matrix for $T5_{base}$ model.

Actual\Predict	**Standard**	**Non-standard**
Standard	1,129	251
Non-standard	214	1,029

c Confusion matrix for the proposed model.

Actual\Predict	**Standard**	**Non-standard**
Standard	1,229	151
Non-standard	353	855

d Confusion matrix for GPT-4.

Table 4.5: Confusion matrices for Japanese Twitter experiment.

a standard instance as non-standard, while the bottom left cell represents the false-negative cases, where the model failed to detect a non-standard instance, incorrectly labeling it as standard. WORD2VEC model, shown as (a) in Table 4.5, produces a high rate of false positives, indicating frequent misclassification of standard instances as non-standard. $T5_{base}$ model, shown as (b), achieves the lowest false-positive rate, but the model struggles with detecting non-standard instances, resulting in the highest rate of false negatives. The proposed model, shown as (c), offers a balanced approach, reducing false negatives more effectively than the other models, which suggests it is particularly strong at identifying non-standard instances. Although the proposed model does not achieve the lowest number of false positives, its overall performance in minimizing misclassifications makes it the most reliable model for distinguishing between standard and non-standard word usages in Japanese Twitter data. GPT-4, shown as (d), tends not to produce the non-standard label as frequently as other models, yet it achieves the best precision for this label. This indicates that the model only detects non-standard usage when it has high confidence in Japanese texts.

Failure Type	Sentences
False-Positive	(a) それとも私が**藁**をもすがりたい男に見えたのだろうか。 (b) サタンメフィスト、**ピザ**って 10 回言ってー メフィ私はパスタ派なんですが サタンそっかぁー (c) 北海道民だからって特別**蟹**が好きじゃないし、たくさん食べたいとも思わないし
False-Negative	(d) そう言えば松宮さんダン戦Wの**円盤**箱で持ってるけど受験終わった一気に見ようかな (e) 最近高良健吾の**沼**にはまってるので中村一太が高良健吾に見えて仕方なかった (f) ファボった人でオフ会やっても麻雀すら出来なさそうで**草**生えるｗｗ

Table 4.6: Example false-positive and false-negative cases by the proposed method. **Bold** indicates the target word.

Table 4.6 shows example error cases produced by the proposed model for each failure category. The top three word types for misclassification are "藁", "ピザ", and "蟹" for false-positive cases, and "円盤", "沼", and "草" for false-negative cases. These top three word types account for 46.9% of the total false-positive errors and 28.7% of the total false-negative errors. We observed a false-positive pattern in narrative cases, such as case (b) in Table 4.6, where the model tends to classify a sentence as non-standard if it contains dialog between characters or humans. However, this pattern did not represent a major cause of the overall false-positive cases. For false-negative errors, the model tends to ignore phrasal slang that is often used on the web, such as in case (e) "沼にはまる" (becoming enthusiastic about something)" or case (f) "草生える" (indicating laughter in a similar fashion to "lol", laughing out loud, in English). To prevent such failures, we need to carefully choose the corpora for the pre-trained model and pseudo-label training to avoid including examples that could introduce a bias toward net slang in the masked language model and the usage classifier.

4.4 Experiments on English Social Media Dataset

In the previous experiment, we addressed the word usage classification task using a pre-annotated dataset. For practical applications, we extend this task into a non-standard usage discovery task by applying the classification over unannotated corpora to all possible candidate words, selecting sentences and their associated target words with the highest scores for the non-standard label. In this experiment, we utilize Reddit[10] as a data source to validate our method for the discovery task through crowd-sourced evaluation.

[10]https://www.reddit.com

Subreddit	# Users	# Sentences
r/hiphopheads	51,372	1,003,320
r/4chan	42,765	238,228
r/teenagers	35,373	984,618
r/australia	24,636	1,033,833
r/ireland	10,258	303,885

Table 4.7: Selected subreddits and its statistics.

4.4.1 Non-standard Word Usages in English

Data Collection

This section outlines the detailed data creation process. We collect sentences from Reddit posts during 2014 for annotation, and run three kinds of models, WORD2VEC, $T5_{base}$ and our proposed model, over the candidate posts. After aggregating detected non-standard usages obtained by those methods, we perform human annotation to the results via crowd-sourcing and regard annotated sentences as final dataset. Reddit is a social media platform and has various communities called subreddits which are dedicated to a particular topic. We assume that each subreddit has its own slang or jargon related to the community, which gives us some non-standard usages of certain words. The processes involved in this procedure will be explained in detail below.

We first select five subreddits for our experiments as shown in Table 4.7. Based on Del Tredici and Fernández (2018), given that "small-to-medium-sized" communities are more prone to lexical innovations than larger communities with 15 million users, we selected subreddits that fits this size. In addition to a niche community, r/hiphopheads, we also include subreddits of a more general nature such as r/4chan or r/teenagers to add some diversity, and r/ireland or r/australia to investigate regional variations of English.

We next create a set of word tokens (and their associated sentences) for a subreddit where the words are used as a noun, as in the previous experiment. We use the sentence chunker from nltk (Bird, 2006), lowercase all words and perform part-of-speech tagging with TreeTagger (Schmid, 1999).

We filter out nouns that do not occur in COCA[11], the corpus of contemporary American

[11] https://www.english-corpora.org/coca

English, as we focus on unusual usage of an in-vocabulary word. Similar to our approach with the Japanese dataset, we also filter out words that appear fewer than 100 times in COCA.

We run baselines methods, WORD2VEC and $T5_{base}$ and our proposed model over the set of words and sentences, and collect non-standard usages of word type that are predicted by the models. Note that a word type is considered non-standard by a model only if it has at least 10 non-standard usages by different users. As a result, we have 40–1200 candidate word types per subreddits.

Lastly we randomly select five different word types from each subreddits per method, resulting in a total of 15 word types. Using these 15 word types, we extract the 10 most non-standard usages/sentences detected by each method. In total, we select 75 word types, yielding 750 usages per method and resulting in a dataset of 2,250 sentences.

Human Annotation

We perform human annotation of the collected data using Amazon Mechanical Turk (Mturk).[12] In this annotation, each worker is asked whether the given target word in a sentence is used in a standard or non-standard way, on a scale of 1 to 5. In more detail, workers are asked to mark non-standard usages as 5 ('Very Unusual') or 4 ('Unusual') for the given target word, and 2 ('Natural') or 1 ('Very Natural') otherwise. The slider is set to a default position of 3, which is not used in the annotation task, and the system automatically blocks submission if the worker does not move the slider towards either 1 or 5. A snapshot of the annotation screen is shown in Figure 4.4.

In this annotation, workers are asked to annotate 32 sentences in a HIT (= annotation session). We include two quality control items in each set, in the form of one sentence extracted from English Wikipedia (which workers needed to annotate as 2 or 1) and one sentence from English Wikipedia where a noun was randomly replaced with the target word (which workers needed to annotate as 4 or 5). We employed several filtering strategies to ensure higher quality from workers: a HIT approval rate greater than 95%, more than 100 accepted HITs, and United States as location. We rejected HITs where one of the quality control items was annotated other than expected[13]. In total, 273 workers participated in the combined annotation task, and five different workers annotated each

[12] https://www.mturk.com

[13] We rejected HITs that failed both control questions for the first 47% of the tasks. However, this restriction yielded a 73% rejection rate, making it difficult to collect sufficient data. Hence, we relaxed the conditions thereafter.

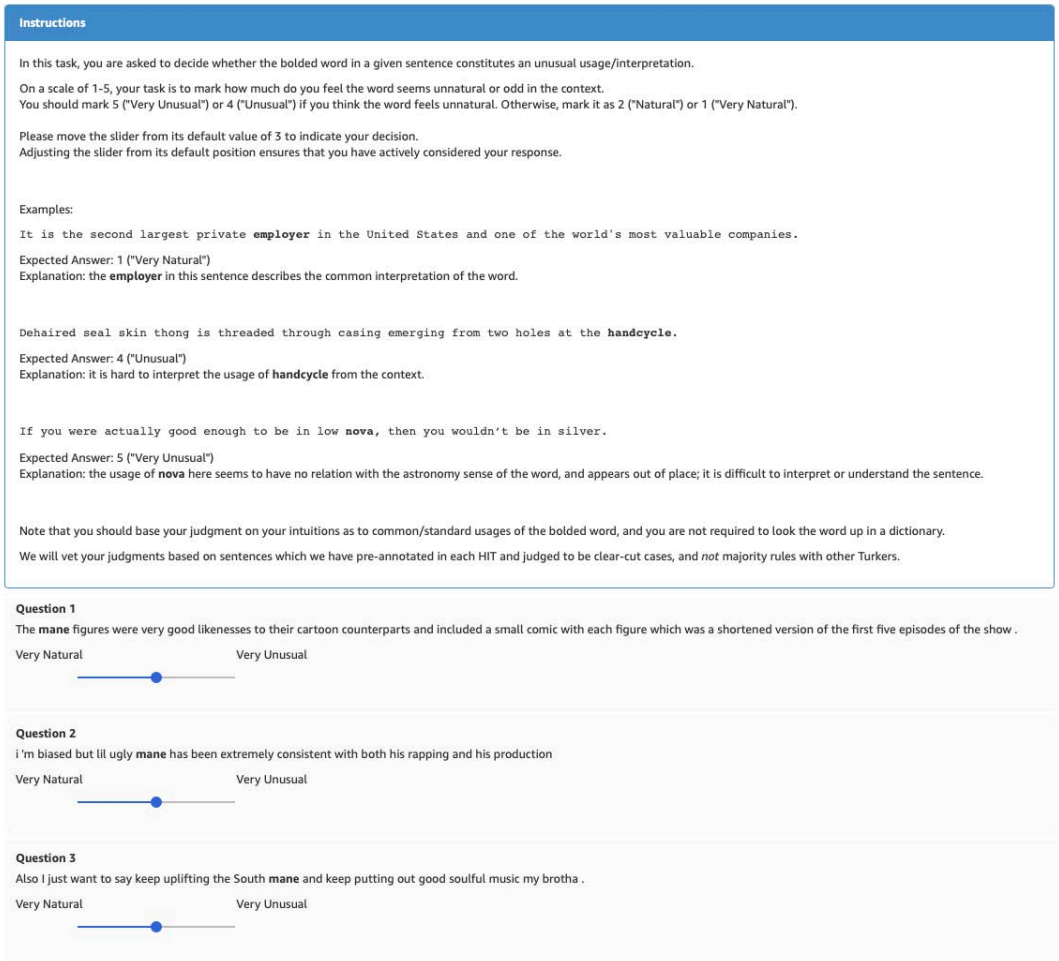

Figure 4.4: Example MTurk snapshot. Workers are required to move the slider to submit their responses.

sentence.[14] The average time for workers to complete a HIT was 37.12 minutes, which translates to approximately 1.16 minutes per question. To mitigate individual variation in the annotated labels, we binarized the labels by converting 5 ('Very Unusual') or 4 ('Unusual') into a non-standard label and 2 ('Natural') or 1 ('Very Natural') into a standard label as a post-process. We then assessed inter-annotator agreement using Fleiss' Kappa (Fleiss et al., 1971) to investigate the quality. The agreement study yielded a score of 0.1129, which falls into the category of slight agreement (Landis and Koch, 1977), indicating that the task involves a certain level of subjectivity among annotators.

[14]Note that we allocate several workers per instance to alleviate individual variation and investigate the majority of the understanding for the term as the annotation task depends on the subjectivity or background knowledge of the workers.

Subreddit	#Standard	#Non-standard
r/4chan	105	104
r/hiphopheads	220	40
r/ireland	224	32
r/teenagers	256	30
r/australia	271	28
Total	1,076	234

Table 4.8: The details of the labels for the created dataset.

Example non-standard word usages	
Subreddit	*Sentences*
r/4chan	(a) He shoved the sandwich in her face and said skedaddle , it's actually **alpha** as ****.
r/hiphopheads	(b) lame friends who know nothing about hip hop lol... you cant let ppl rag on redman or **method**.
r/ireland	(c) What did McWilliams have to do with the **tiger** , was future shock not a warning?
r/teenagers	(d) taking some physics and calculus this semester and a couple **gen** ends will do the same over summer and then start my engineering courses which are formatted differently at Purdue
r/australia	(e) Nice shot , great technique , so did u flip a **grad** and to help control the city ambient light ?

Table 4.9: Example sentences that are annotated as non-standard by at least 4 workers. **Bold** indicates the target word. Profanities have been redacted. Explanation of each usage: (a) a dominant or assertive male; (b) *Method Man*, a famous rapper in United States; (c) an abbreviation of general as in General Education; and (d) *Celtic Tiger*, the rapid economic growth in Ireland from the 19th–20th century; and (e) an abbreviation of graduated filter.

For creating the final dataset, we extracted sentences where at least 4 workers agreed on either a standard or non-standard label and discarded the others. It is important to note that this process is a trade-off between the quantity and quality of the dataset; we discarded 940 examples that potentially contains gold labels. However, given the fact that the annotation task depends on the subjectivity of the worker, we decided to prioritize quality over quantity. As a result, 1,076 examples were labeled as standard, and 234 examples were labeled as non-standard. Table 4.8 shows detailed annotation results. As illustrated in the table, r/4chan contains the most non-standard usages in the dataset, whereas r/australia reflects more common usages.

Observations

Example non-standard word usages are shown in Table 4.9. As the examples demonstrate, each usage somewhat reflects the culture of a particular community; r/4chan

and r/teenagers tend to include more slang, while r/hiphopheads is related to a specific music genre. We found that *gen* in case (d) in Table 4.9 used in r/teenagers, refers to "general", as in General Education, often abbreviated as "Gen Ed" in the context of curriculum or courses in a university. We also found the term *tiger* in case (e) is often used to refer to *Celtic Tiger* in r/ireland that reflects its geographical information about the community. Contrary to our expectations, however, we did not observe many dialectal usages in r/australia and r/ireland communities.

Building on the Japanese examples, Table 4.9 highlights parallel trends in English social media. For example, *method* signifies "Method Man", just as "尼 (literally nun)" can mean Amazon.com in Japanese. Meanwhile, terms like *gen* or *alpha* mirror the new or subcultural senses that Japanese non-standard word usages frequently exhibited in Chapter 3. In both languages, context and insider familiarity govern these non-standard usages, underscoring the need for flexible, culture-aware NLP approaches.

4.4.2 Model Details

For the English experiments, we implemented the proposed method using English resources. We trained word embeddings on the COCA and trained the classifier using the pre-trained English RoBERTa$_{\text{base}}$[15] on the ukWaC corpus (Ferraresi et al., 2008). Similar to the Japanese Twitter experiment, we generated one million standard and pseudo-examples from the corpus, along with 1,000 sentences for validation data. The same hyperparameters as those used in the Japanese experiment were applied, and we selected the model based on its best performance on the validation dataset. The prompt used in GPT-4 is shown in Table 4.10.

For WORD2VEC baseline, we utilized the embeddings trained in the aforementioned process with a threshold of -0.02 determined based on the validation dataset described previously. Unlike the previous experiment on Japanese dataset, where we determined the value based on 10-fold cross-validation over test data, real-world applications like this English experiment require the model to have a single threshold applicable to all sentences. We also employed the pre-trained English T5$_{\text{base}}$ model[16] and followed the

[15]https://huggingface.co/FacebookAI/roberta-base
[16]https://huggingface.co/google-t5/t5-base

same training strategy as the proposed model[17].

4.4.3 Evaluation Metrics

Using the newly annotated dataset, we investigate the usage classification performance of the baselines as well as our proposed model. As in the previous experiment, we used accuracy, precision, recall, F1 score, and average precision to evaluate the rankings, with results from GPT-4 provided as a reference.

[17]Note that the pre-training corpora for $\text{RoBERTa}_{\text{base}}$ and T5_{base} differ, complicating direct comparisons between the models. Specifically, $\text{RoBERTa}_{\text{base}}$ is trained on BookCorpus, Wikipedia, CC-News, OpenWebText, and Stories, while T5_{base} uses the Colossal Clean Crawled Corpus (C4) and Wikipedia. These differences in training data likely result in distinct linguistic patterns and biases, impacting their performance on downstream tasks.

Prompt
INSTRUCTION: In this task, you are asked to decide whether the target word in a given sentence constitutes an unusual usage/interpretation. If the usage is standard or natural, respond with ”YES”; if it is not, respond with ”NO.” ### INPUT: SENTENCE: They threatened to nuke the enemy city. TARGET WORD: nuke ### OUTPUT: YES ### INPUT: SENTENCE: I had to nuke the entire server to get rid of the virus. TARGET WORD: nuke ### OUTPUT: NO ### INPUT: SENTENCE: The tank rolled through the streets, demonstrating its power. TARGET WORD: tank ### OUTPUT: YES ### INPUT: SENTENCE: In our raid, I play as the tank to absorb the boss’s attacks. TARGET WORD: tank ### OUTPUT: NO ### INPUT: 文章: {*sentence*} 単語: {*target_word*} ### OUTPUT:

Table 4.10: English prompt used for GPT-4.

4.4.4 Results

Table 4.11 shows the results on the English Reddit dataset. Our proposed model outperforms baselines model on all the major evaluation metrics. The absolute improvement

Model	Acuuracy	Precision	Recall	F1	Average Precision
WORD2VEC (SGNS IN-OUT)	.342†	.156	**.611**	.249	.141†
$T5_{base}$	.669†	.189	.261	.219	.180†
Proposed Method	**.762**	**.306**	.256	**.279**	**.245**
GPT-4.0	.789	.434	.594	.502	-

Table 4.11: Experimental results on the English Reddit Dataset. **Bold** indicates the best score. A dagger symbol (†) indicates there is a statistical significance ($p < .01$) against proposed method.

between the second-best baseline and the proposed model was 9.3, 3.0 and 5.5 points for accuracy, F1 score and average precision respectively. We observed a statistical significance with a level of $p < .01$ for accuracy and average precision based on permutation test. There is no statistical significance observed for the F1 score.

As we can see from Table 4.11, WORD2VEC suffers from overall classification performance. One reason for this is that the WORD2VEC method requires a global threshold for decision-making. In WORD2VEC method, it is likely that a single global value cannot effectively work for all examples, and we might need to fine-tune the threshold based on the specific word. In contrast, our proposed pseudo-data based method does not require any threshold tuning for classification achieving the best accuracy and proving to be effective for this task. Compared to $T5_{base}$, our proposed method achieves higher evaluation result for across all metrics, highlighting the importance of pre-trained masked language models for distinguishing word usages. In contrast to the experiment on the Japanese dataset, GPT-4 achieves the best performance across all the metrics. Although the technical report (OpenAI, 2024) does not explicitly mention about the training data for GPT-4, the model demonstrates a stronger capability in handling English texts than other languages as reported by its higher performance on language understanding tasks such as GLEU (Wang et al., 2019). This tendency for higher performance on English also applies to our word usage classification task. However, as we mentioned in Section 4.3.3, using LLMs is not scalable as the current prompt only produces a binary label for the given input string, making it difficult to determine which specific usage is more likely to be non-standard. In addition, inference with GPT-4 is financially expensive and

WORD2VEC	$T5_{base}$	Proposed	GPT-4	*Sentences*
✓	✓	✓	✓	... never having done it, because your post gave me **cancer** anyway.
✓	✓	✓	✓	not *that* into **gen** lll to be honest but heck yeah i'm getting Omega Ruby...
		✓	✓	... end up retiring in 2022 and will make **ton** more profit in real estate...
			✓	I think the game was clever and interesting and made good use of the **medium** ...
				If you want you can PM your **origin /** steam username and we can play something.
				For me it's when my friend will **text** me something like "hey dude ...

Table 4.12: Example sentences and corresponding model predictions. **Bold** indicates the target word. A checkmark (✓) denotes a correct model prediction.

not feasible for large text streams, such as those on Twitter or Reddit.[18] As noted in the previous experiment, prompt engineering to boost the performance or teach the model to produce a scalar value is out of the scope of this paper.

Table 4.12 shows a quality comparison of each method. All models made correct predictions for the first two sentences. Interestingly, these sentences contain metaphorical expressions, such as "gave me cancer" and "mad as hell". These phrases are not literal but are used to convey strong emotions or negative experiences. Despite the figurative language, all models successfully captured the intended meaning, showing their ability to handle metaphorical usage. The proposed method and GPT-4 correctly predict certain challenging sentences, even for the typo "ton", which was originally meant to be "ten". None of the methods correctly predicted non-standard usages[19] when the literal meaning also fits the context. For instance, in Table 4.12, the word "origin" refers to a gaming platform, yet its literal meaning fits the sentence contextually. Similarly, the word "text", commonly used in digital communication to mean sending a message, was not correctly identified despite its increasing prevalence in internet language. Developing more robust models to address these cases is left for future work.

Table 4.13 shows the evaluation metrics for each subreddit by the proposed model. Overall, the model performs inconsistently across subreddits, with `r/4chan` showing

[18] As of August 2024, it would approximately cost over $50,000 just to apply the classification to the examples that our model processed to create the dataset.

[19] Note that there are cases where the model's output is standard, yet the annotation label is non-standard. This occurs because we collect the top 10 usages sorted by the scores for non-standard labels; however, not all usages are classified as non-standard if the score for a usage does not meet the threshold.

Subreddit	Acuuracy	Precision	Recall	F1	Average Precision
r/4chan	.541	**.643**	.173	.273	**.618**
r/hiphopheads	.788	.377	**.575**	**.455**	.389
r/ireland	.801	.194	.188	.290	.164
r/teenagers	.808	.143	.167	.154	.115
r/australia	**.823**	.195	.286	.232	.179

Table 4.13: Evaluation metrics for each subreddit by the proposed method.

Example detected non-standard word usages	
Subreddit	*Sentences*
r/4chan	(a) You are **alpha** , you will ignore it, or fight it with the counter, the pride in what you are.
r/4chan	(b) do please go ahead and eat that **cancer** right up with the rest of the newfriends, friend
r/ireland	(c) EDIT : Should say that I 'm on **meteor** now and have no such problems.
r/teenagers	(d) **text** your mom " I use your toothbrush to apply the toothpaste to ****, to get high"
r/4chan	(e) ... you have those I mean there has to be at least a couple bro **tier** who hang around there.
r/australia	(f) ... a group against the hostile and unlawful treatment of asylum seekers ... thats 4chan **tier**
r/hiphopheads	(g) ... angsty high school **tier** lyrics get in the way of some otherwise dope production.
r/4chan	(h) I am trained in **gorilla** warfare and Im the top sniper in the entire US armed forces.
r/hiphopheads	(f) ... how do you have so little comment karma **eve** though u have been here for a year.

Table 4.14: Example detected non-standard usages. **Bold** indicates the target word. Longer sentences have been shortened and profanities have been redacted. Explanation of each usage: (a) a dominant or assertive male; (b) a toxic person or behaviour; (c) a mobile network company in Ireland; (d) sending a message; (e-g) a slang refers to ranking something or someone by level or quality; (h) a typo for "guerrilla"; and (f) a typo for "even".

strong precision but weak recall, r/hiphopheads offering a more balanced performance, and r/ireland, r/teenagers, and r/australia demonstrates high accuracy but low precision and recall. One possible reason for this tendency in accuracy is the imbalance in the dataset containing non-standard usages; r/4chan and r/hiphopheads contain the most non-standard labels, with 104 and 40 non-standard usages respectively, while r/australia and r/teenagers contain the least, with 28 and 30 non-standard usages respectively, resulting in higher accuracy for subreddits with more standard usages.

Table 4.14 illustrates example non-standard usages detected by the proposed model. As shown in case (a-d) in Table 4.14, the proposed model successfully detects net slangs

Actual\Predict	**Standard**	**Non-standard**
Standard	305	771
Non-standard	91	143

a Confusion matrix for WORD2VEC (SGNS IN-OUT).

Actual\Predict	**Standard**	**Non-standard**
Standard	815	261
Non-standard	173	61

b Confusion matrix for $T5_{base}$ model.

Actual\Predict	**Standard**	**Non-standard**
Standard	940	136
Non-standard	174	60

c Confusion matrix for the proposed model.

Actual\Predict	**Standard**	**Non-standard**
Standard	895	181
Non-standard	95	139

d Confusion matrix for GPT-4.

Table 4.15: Confusion matrices for English Reddit experiment.

or non-standard usages across different subreddits. Interestingly, as shown in cases (e-g), the model detects the same word *tier* as non-standard, which may indicate a signal that the term is likely a more general slang term rather than community-specific jargon (Del Tredici and Fernández, 2018). Additionally, as shown in cases (h-f), the model is able to detect typos labeling them as non-standard suggesting that it could potentially be applied to grammatical error correction at the lexical level (Ng et al., 2014; Kiyono et al., 2019; Kaneko et al., 2020) or data cleaning for further NLP applications (Saito et al., 2014, 2017; Bolding et al., 2023).

4.4.5 Error Analysis

Similar to the previous experiment with the Twitter dataset, we analyzed the error cases using confusion metrics and conducted an example error case study.

Table 4.15 shows confusion matrix for each model to analyze classification errors. As noted in Section 4.3.5, we regard positive as non-standard usages and negative as

standard usages, and the top right cell in Table 4.15 represents the false-positive cases (incorrect detection), while the bottom left cell represents the false-negative cases (missed detection). WORD2VEC model, shown as (a) in Table 4.15, achieves the greatest number of correct detections for the non-standard label, but it also has an increasingly high rate of false positives, which lowers the overall metrics. This may be related to a threshold determined by the validation set, which reveals a discrepancy between the synthetic dataset and actual texts. $T5_{base}$ model, shown as (b), produces a similar number of false-negatives compared to the proposed model but has a higher rate of false-positives. The proposed model, shown as (c), demonstrates a strong performance in distinguishing between standard and non-standard labels. Compared to WORD2VEC model and the $T5_{base}$ model, the proposed model achieves the highest precision for standard label detection, with fewer misclassifications as non-standard. However, similar to the $T5_{base}$ model, it shows a relatively high number of false negatives for the non-standard label. As a reference, GPT-4 shown as (d) demonstrates a strong capability in correctly identifying both standard and non-standard labels, with relatively lower rates of false positives and false negatives. This indicates that GPT-4 provides a more accurate and reliable classification performance, making it a robust choice for handling the variability in the dataset depending on the language. The overall performance suggests that while the proposed model excels at identifying standard labels, it faces difficulties in accurately detecting non-standard labels, similar to the other models compared. The performance of GPT-4 on the English dataset is not as strong as on the Japanese dataset, and these underwhelming outcomes further highlight the complexity of the task, including challenges posed by annotation errors from crowd-sourced evaluations. We will address these issues in the following paragraph, where we conduct an example-wise error analysis that includes a discussion of annotator errors, supported by explanations for each case.

Table 4.16 shows example error cases produced by the proposed model for each failure category. As we can see from the results, the false-positive cases include a literal usage of *method* as in case (a) in Table 4.16 or *revenue* to mean a financial institution as in case (b). However, it also contains annotation errors, such as *mane* in case (c) being used to mean an American rapper "Gucchi Mane", or *guise* in case (d) meaning its homophone "guys". For false-negative cases, the model fails to detect non-standard or contextually nuanced usage of certain words. For instance, the model incorrectly treats *hell* in case (e) as a standard usage rather than recognizing it as a name, despite the use of "his" as a reference. Likewise, the model fails to identify *alpha* in case (f)

Failure Type	Sentences
False-Positive	(a) ... with extreme attention to detail and great execution, much like a director or **method** actor. (b) ... got a scam email supposedly from irish **revenue** saying they owed me and fill in my details. (c) ... really really happy phase and that was on repeat it's just a feel good song to me **mane**. (d) **guise** ireland better than all of u , here s a lovely shot of our artistic culture.
False-Negative	(e) I think **hell** want to ride the press wave of squashing his beef with the game (f) He shoved the sandwich in her face and said skedaddle, it's actually **alpha** as ****. (g) Thanks , I should really pick up that "**vocabulary**" everyone's talking about. (h) Going to Asia for a **height** problem is pretty pathetic but considering how unwanted I feel..

Table 4.16: Example false-positive and false-negative cases by the proposed method. **Bold** indicates the target word. Longer sentences have been shortened and profanities have been redacted.

as part of a slang expression, highlighting the difficulty the model faces in handling context-specific colloquial usage. Furthermore, similar to the false positive cases, we have identified annotation errors. Specifically, MTurk workers tend to label usages as non-standard when the target word is used in an interrogative form within a sentence (case (g)) or when the overall meaning of the sentence is unclear or controversial (case (h)). To prevent such undesired annotation errors in the future, more detailed instructions should be provided on MTurk, explicitly mentioning cases to avoid along with relevant examples.

2Finally, we outline future avenues of error analysis that have not yet been addressed in this dissertation. Real-world applications may require different threshold settings depending on their objectives. For instance, a stricter threshold can be used to increase precision, while a more lenient threshold can ensure broader coverage. In the proposed method, decisions are made by a binary classifier that treats any label exceeding a 50% probability as positive. For example, by designating cases with a predicted probability of 70% or higher as "non-standard", one can configure a high-precision setting similar to the first method. In practical applications, an important consideration is how much recall can be maintained at a given level of precision, and this remains a topic for future research.

4.5 Discussion

In this study, we were able to detect non-standard word usages from unlabeled corpora and analyze these examples in detail. The following discussion highlights limitations of

the current approach and outlines possible improvements for this task.

Enhancement of human evaluations: According to our human annotation tasks on the usages detected by each method, crowd-workers sometimes struggled with ambiguity in determining whether certain examples were standard or non-standard. Specifically, we found instances where both our proposed method and GPT-4 labeled certain examples incorrectly, where those examples turned out to be annotation errors. To address this issue, we could add more stringent qualifications for crowd-workers, such as geographic or educational background requirements, refine the task instructions, increase the number of quality control questions, or employ multiple annotators for cross-validation.

More accurate detection of non-standard word usages: Although we found that the proposed methods outperformed LLM-based methods like GPT-4 or Swallow for Japanese, we also observed that GPT-4 performs quite well for English. As discussed in Chapter 1, applying LLM-based methods to massive text streams (e.g., those found on X or Reddit) poses both computational and financial challenges. To address these concerns while still pursuing high accuracy in this task, a two-stage hybrid detection approach similar to Huang et al. (2023) can be adopted, which consists of (1) a candidate extraction stage and (2) a refinement stage. In the first stage, candidates would be extracted in advance by an automated non-standard detection method, such as those proposed in the first or second study. Experimental results in the second study show that the word embedding-based method proposed in the first study yields higher recall and thus would be more suitable not to ignore ground-truth non-standard word usages in the second refinement stage. Alternatively, we can apply ensemble-technique (Shazeer et al., 2017; Huynh et al., 2020) to ensure the coverage. Once candidates are narrowed down in the first stage, the second stage can rely on an LLM-based method for the final decision. Further prompt-engineering techniques, such as Chain-of-Thought (Wei et al., 2022), Self-Discover (Zhou et al., 2024), or Auto-Evolve (Aswani et al., 2024), may also enhance the accuracy of LLMs, that are beyond the scope of this thesis. Ultimately, this two-stage approach could mitigate the cost and scalability issues that are originally associated with LLM-based solution for this task.

More scalability in language expansion: An important feature of the proposed approach is its relative language independence, given that it primarily relies on distributional representations of words learned from unlabeled corpora. However, the methodology still depends on having a pre-trained language model available for each target language. One viable approach to further minimize language dependence is to adopt

multilingual pre-trained models, such as XLM-R (Conneau et al., 2020), a multilingual variant of RoBERTa that can handle multiple languages in a single framework. Despite this advantage, multilingual approaches can introduce challenges related to vocabulary overlap and representational interference across languages (Rust et al., 2021), potentially lowering performance for language-specific nuances. Moreover, Zhao and Aletras (2024) found XLM-R less faithful than monolingual RoBERTa, and the larger the multilingual model, the less faithful its rationales became compared to its monolingual counterpart. This concern may be critical in assessing whether a multilingual or monolingual approach is more suitable for non-standard word usage detection.

4.6 Conclusion for this Chapter

In this work, we introduced a new English Reddit dataset focusing on non-standard word usage and proposed a RoBERTa-based model for detecting such usages in social media text. To deal with the lack of annotated data for non-standard usage detection, we incorporate pseudo-example learning, where we train the model over synthetic examples generated from a corpus. Experimental results on a Japanese X dataset and English Reddit dataset show that our proposed method achieves better performance across a range of evaluation metrics and different languages than baseline methods. Our experiments show that fine-tuning LM head and incorporating Skip-gram information are essential for improving model performance, enabling better adaptation to non-standard usages. These findings underscore the value of combining different sources of pre-trained knowledge to enhance the model ability to distinguish non-standard usage patterns. Although we focus on noun cases, the proposed model is able to handle other parts of speech or even phrases, which we leave for future work.

Chapter 5

Conclusion

5.1 Conclusion

This thesis investigated non-standard word usages on multilingual social media and proposed language-agnostic, annotation-free methods for effectively detecting them.

In the first work, we introduced a newly created Japanese evaluation dataset from X, featuring example usages annotated by domain experts, and proposed a lightweight word embedding approach for detecting non-standard word usages. The proposed method is based on Skip-gram and combined both $\mathbf{v}^{IN}$ and $\mathbf{v}^{OUT}$ word vectors obtained through its representation learning algorithm. Specifically, we computed a weighted average of the sigmoid-transformed dot products between the target and surrounding word vectors as a usage standardness score, with higher scores indicating standard usage and lower scores signifying non-standard usage. In the proposed method, the final classification of the usage was determined by a threshold. We found that incorporating $\mathbf{v}^{OUT}$ yields higher performance, suggesting that output-side word vectors are especially useful for referencing both the target word and its surrounding words. Also weighting surrounding words according to their distance from the target further improved performance, indicating that nearer words provide more critical cues for non-standard word usage detection.

In the second work, we proposed a masked language model-based method that eliminates the need for a global threshold and incorporates broader contextual information to overcome the limitations of the first approach. To evaluate this method, we introduced an English Reddit dataset annotated by crowd-workers. The proposed method is based on masked language modeling with a RoBERTa-driven word usage classifier, leveraging capabilities of the pre-trained model to accurately detect non-standard word usages across diverse linguistic settings. To mitigate the scarcity of labeled data for non-

standard usages, we employed pseudo-example learning, whereby synthetic training examples are generated from unlabeled corpora. Our multilingual experiments revealed that this approach achieves superior performance across various metrics both in English and Japanese compared with baseline methods. Additionally, fine-tuning the LM head and integrating the word embedding information, introduced in Chapter 3, led to greater adaptability in detecting non-standard word usages. These findings show the benefit of fusing diverse pre-trained knowledge to better capture evolving, informal language in real-world social media contexts.

5.2 Future Work

Finally, we present possible future directions for this line of research.

Extension of the research target: Although our work centered on nouns, neither of the proposed methods in Chapters 3 and 4 depends on a specific part of speech. Investigating all word classes, rather than limiting the scope to nouns, could provide a clearer view of the practical landscape of non-standard word usages. In addition, extending these methods to other languages remains a key consideration. While we initially focused on Japanese and English, the RoBERTa model, employed in the second study, supports over 100 languages, which naturally allows the proposed method to be applied in broader cross-lingual settings. By investigating these additional languages, researchers could gain a deeper, comparative perspective on non-standard word usages.

Potential NLP applications by extending this work: In analyzing non-standard word usages, understanding their underlying meanings is as crucial as extracting examples, since it broadens our comprehension of these expressions. Building on the findings of this thesis, several avenues of NLP research can be pursued to deepen our understanding of non-standard word usages. Based on our observations in Chapters 3 and 4, non-standard word usages typically refer to (1) new knowledge or (2) existing knowledge. In both cases, definition modeling (Noraset et al., 2017; Huang et al., 2021, 2022; Segonne and Mickus, 2023) will help us interpret and introduce a new sense of the term. We believe that these insights can be incorporated into knowledge bases like BabelNet (Navigli and Ponzetto, 2012) to expand NLP capabilities for addressing non-standard expressions. In turn, applications that integrate enriched knowledge for downstream tasks such as information retrieval (Moro et al., 2014; Feng et al., 2020) or machine translation (Pham et al., 2018; Pu et al., 2018) can more reliably process unconventional terms.

Acknowledgements

Although this dissertation is written in English, I have chosen to present the following acknowledgements in my native language, Japanese, to convey my heartfelt gratitude without losing any nuances through translation.

本学位論文は著者が東京工業大学工学院情報通信系情報通信コースに在学中の研究成果をまとめたものです。本論文の審査を引き受けてくださった、産業技術総合研究所高村大也氏、東京科学大学篠崎隆宏教授、鈴木賢治教授、中山実教授、船越孝太郎准教授、に感謝申し上げます。本研究を遂行するにあたり、多くの方のご支援、ご指導をいただきました。

まず、指導教員である奥村学教授には、修士課程を含め 9 年にもわたってご指導いただきました。在学中は新型コロナウイルス感染症の蔓延によるさまざまな困難に見舞われましたが、丁寧なご指導とご鞭撻を頂戴いたしました。さらに、博士後期課程を満期退学した後、突然研究を再開したいと申し出た際にも、快く受け入れてくださいました。お忙しい中、定期的にお時間を割いて熱心にご助言いただけたからこそ、最後まで研究を進めることができました。ここに深く感謝を申し上げます。

本研究を遂行するにあたり、修士課程・博士課程の指導教員を務めてくださった産業技術総合研究所の高村大也氏には、多大なるご指導とご鞭撻を賜りました。博士課程在学中には、留学やインターンシップへ参加するなど、落ち着きのない私をおおらかに支えていただき、誠に感謝しております。英語論文の執筆時にはご尽力いただき、論文執筆のいろはを学ぶことができました。深く感謝いたします。

また、MBZUAI 所属の Timothy Baldwin 教授ならびにメルボルン大学の Jey Han Lau 准教授には、2018〜2019 年のメルボルン大学留学中に熱心なご指導を頂戴いたしました。Tim 先生には、修士課程 2 年次のころに私から留学の相談を持ちかけた際、英語力に不安があるにもかかわらず快く受け入れていただきました。この留学は私の人生における大きな転機となり、おふたりを通じて研究のみならず多岐にわたる知見を得ることができました。お力添えに深く感謝申し上げます。

船越孝太郎准教授ならびに奈良先端科学技術大学院大学の上垣外英剛准教授には、博士課程在学中に議論を通じて多くのご指導と助言を賜りました。留学中やコロナ禍の影響で直接お会いする機会は限られていましたが、両先生からの建設的なご指摘により、研究を深めることができました。誠にありがとうございます。

本研究の着想に至るにあたり、修士課程在学中にご指導いただいた名古屋大学の笹野遼平准教授にも、多大なるご助力をいただきました。研究の右も左も分からない私に対し、丁寧かつ細やかなご指導を賜りましたことに深く感謝申し上げます。在籍期間が重なっていたのは短い間でしたが、私が研究に興味を持つようになったきっかけは、間違いなく笹野さんとの共同研究によるものです。この体験は、研究の楽しさを知る大きな転機となりました。ここに改めて御礼申し上げます。

研究室の皆様にも、大変お世話になりました。研究室秘書の飯山信子氏には、RA、留学、出張申請、学位申請など、あらゆる事務手続きで多大なご支援をいただき、心より感謝申し上げます。計算機の管理に尽力してくださった、上垣外准教授、長谷川君、藤田君、小林君をはじめとするアドミンの皆様、そして満期退学後に作業をサポートしてくださった前川さん、家田さん、Kwon 君、Ma さんをはじめとするアドミン・総務の皆様にも、大変お世話になりました。多大なるご支援に御礼申し上げます。

最後に、これまで私を応援してくれた家族、そして献身的に支えてくれた妻に、心より感謝申し上げます。

References

[1] Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv*, arXiv:1803.08375.

[2] Kushal Arora, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. 2022. Director: Generator-classifiers for supervised language modeling. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 512–526, Online only. Association for Computational Linguistics.

[3] Krishna Aswani, Huilin Lu, Pranav Patankar, Priya Dhalwani, Xue Tan, Jayant Ganeshmohan, and Simon Lacasse. 2024. Auto-evolve: Enhancing large language model's performance via self-reasoning framework. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13243–13257, Miami, Florida, USA. Association for Computational Linguistics.

[4] AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 33–40, Sydney, Australia. Association for Computational Linguistics.

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

[6] David Bamman, Chris Dyer, and Noah A. Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 828–834, Baltimore, Maryland. Association for Computational Linguistics.

[7] Kobus Barnard, Matthew Johnson, and David Forsyth. 2003. Word sense disambiguation with pictures. In *Proceedings of the HLT-NAACL 2003 Workshop on Learning Word Meaning from Non-Linguistic Data*, pages 1–5.

[8] Fabian Barteld. 2017. Detecting spelling variants in non-standard texts. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 11–22, Valencia, Spain. Association for Computational Linguistics.

[9] Andrei Bejgu, Edoardo Barba, Luigi Procopio, Alberte Fernández-Castro, and Roberto Navigli. 2024. Word sense linking: Disambiguating outside the sandbox. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14332–14347, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

[10] Shane Bergsma, Dekang Lin, and Randy Goebel. 2008. Discriminative learning of selectional preference from unlabeled text. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 59–68, Honolulu, Hawaii. Association for Computational Linguistics.

[11] Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia. Association for Computational Linguistics.

[12] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

[13] Quinten Bolding, Baohao Liao, Brandon Denis, Jun Luo, and Christof Monz. 2023. Ask language model to clean your noisy translation data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3215–3236, Singapore. Association for Computational Linguistics.

[14] Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2024. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4.

[15] Niccolò Campolungo, Tommaso Pasini, Denis Emelin, and Roberto Navigli. 2022. Reducing disambiguation biases in NMT by leveraging explicit word sense information. In *Proceedings of the 2022 Conference of the North American Chapter*

of the Association for Computational Linguistics: Human Language Technologies, pages 4824–4838, Seattle, United States. Association for Computational Linguistics.

[16] Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

[17] 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 the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

[18] Won Ik Cho and Soomin Kim. 2021. Google-trickers, yaminjeongeum, and leetspeak: An empirical taxonomy for intentionally noisy user-generated text. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 56–61, Online. Association for Computational Linguistics.

[19] Minjin Choi, Sunkyung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. 2021. MelBERT: Metaphor detection via contextualized late interaction using metaphorical identification theories. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1763–1773, Online. Association for Computational Linguistics.

[20] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

[21] Paul Cook, Jey Han Lau, Diana McCarthy, and Timothy Baldwin. 2014. Novel word-sense identification. In *Proceedings of COLING 2014, the 25th Interna-*

tional Conference on Computational Linguistics: Technical Papers, pages 1624–1635, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

[22] Peter Crisp, Raymond Gibbs, Alice Deignan, Graham Low, Gerard Steen, Lynne Cameron, Elena Semino, Joe Grady, Alan Cienki, Zoltán Kövecses, and The Group. 2007. Mip: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, 22.

[23] Marco Del Tredici and Raquel Fernández. 2018. The road to success: Assessing the fate of linguistic innovations in online communities. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1591–1603, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

[24] 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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[25] Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, Georgia. Association for Computational Linguistics.

[26] Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed Aly, Beidi Chen, and Carole-Jean Wu. 2024. LayerSkip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12622–12642, Bangkok, Thailand. Association for Computational Linguistics.

[27] Katrin Erk. 2006. Unknown word sense detection as outlier detection. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 128–135, New York City, USA. Association for Computational Linguistics.

[28] Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

[29] Yuwei Fang, Shuohang Wang, Yichong Xu, Ruochen Xu, Siqi Sun, Chenguang Zhu, and Michael Zeng. 2022. Leveraging knowledge in multilingual commonsense reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3237–3246, Dublin, Ireland. Association for Computational Linguistics.

[30] Zhifan Feng, Qi Wang, Wenbin Jiang, Yajuan Lyu, and Yong Zhu. 2020. Knowledge-enhanced named entity disambiguation for short text. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 735–744, Suzhou, China. Association for Computational Linguistics.

[31] Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *In Proceedings of the 4th Web as Corpus Workshop (WAC-4)*.

[32] Jan Fillies and Adrian Paschke. 2024. Simple LLM based approach to counter algospeak. In *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*, pages 136–145, Mexico City, Mexico. Association for Computational Linguistics.

[33] J.L. Fleiss et al. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

[34] Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities.

[35] Jun Gao, Changlong Yu, Wei Wang, Huan Zhao, and Ruifeng Xu. 2022. Mask-then-fill: A flexible and effective data augmentation framework for event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4537–4544, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[36] Spandana Gella, Paul Cook, and Timothy Baldwin. 2014. One sense per tweeter ... and other lexical semantic tales of Twitter. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 215–220, Gothenburg, Sweden. Association for Computational Linguistics.

[37] Spandana Gella, Paul Cook, and Bo Han. 2013. Unsupervised word usage similarity in social media texts. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 248–253, Atlanta, Georgia, USA. Association for Computational Linguistics.

[38] Ariel Gera, Roni Friedman, Ofir Arviv, Chulaka Gunasekara, Benjamin Sznajder, Noam Slonim, and Eyal Shnarch. 2023. The benefits of bad advice: Autocontrastive decoding across model layers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10406–10420, Toronto, Canada. Association for Computational Linguistics.

[39] Waseem Gharbieh, Bhavsar Virendra, and Paul Cook. 2016. A word embedding approach to identifying verb-noun idiomatic combinations. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 112–118.

[40] Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online. Association for Computational Linguistics.

[41] William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Austin, Texas. Association for Computational Linguistics.

[42] William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of EMNLP '16*, pages 595–605.

[43] William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

[44] Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, Oregon, USA. Association for Computational Linguistics.

[45] Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology*, 4.

[46] Viktor Hangya, Qianchu Liu, Dario Stojanovski, Alexander Fraser, and Anna Korhonen. 2021. Improving machine translation of rare and unseen word senses. In *Proceedings of the Sixth Conference on Machine Translation*, pages 614–624, Online. Association for Computational Linguistics.

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

[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[49] Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

[50] Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

[51] Renfen Hu, Shen Li, and Shichen Liang. 2019. Diachronic sense modeling with deep contextualized word embeddings: An ecological view. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3899–3908, Florence, Italy. Association for Computational Linguistics.

[52] Han Huang, Tomoyuki Kajiwara, and Yuki Arase. 2021. Definition modelling for appropriate specificity. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2499–2509, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

[53] Hongzhao Huang, Zhen Wen, Dian Yu, Heng Ji, Yizhou Sun, Jiawei Han, and He Li. 2013. Resolving entity morphs in censored data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1083–1093, Sofia, Bulgaria. Association for Computational Linguistics.

[54] Jie Huang, Hanyin Shao, Kevin Chen-Chuan Chang, Jinjun Xiong, and Wen-mei Hwu. 2022. Understanding jargon: Combining extraction and generation for definition modeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3994–4004, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[55] Peixin Huang, Xiang Zhao, Minghao Hu, Zhen Tan, and Weidong Xiao. 2023. T 2 -NER: A two-stage span-based framework for unified named entity recognition with templates. *Transactions of the Association for Computational Linguistics*, 11:1265–1282.

[56] Huy Duc Huynh, Hang Thi-Thuy Do, Kiet Van Nguyen, and Ngan Thuy-Luu Nguyen. 2020. A simple and efficient ensemble classifier combining multiple neural network models on social media datasets in Vietnamese. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 420–429, Hanoi, Vietnam. Association for Computational Linguistics.

[57] Nancy Ide and Jean Véronis. 1998. Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1–40.

[58] Seiichi Inoue, Mamoru Komachi, Toshinobu Ogiso, Hiroya Takamura, and Daichi Mochihashi. 2022. Infinite SCAN: An infinite model of diachronic semantic

change. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1605–1616, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[59] Shonosuke Ishiwatari, Hiroaki Hayashi, Naoki Yoshinaga, Graham Neubig, Shoetsu Sato, Masashi Toyoda, and Masaru Kitsuregawa. 2019. Learning to describe unknown phrases with local and global contexts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3467–3476, Minneapolis, Minnesota. Association for Computational Linguistics.

[60] Olha Kaminska, Chris Cornelis, and Veronique Hoste. 2021. Nearest neighbour approaches for emotion detection in tweets. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 203–212, Online. Association for Computational Linguistics.

[61] Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.

[62] Daisuke Kawahara and Sadao Kurohashi. 2006. Case frame compilation from the web using highperformance computing. In *Proceedings of LREC '06*, pages 1344–1347.

[63] Casey Kennington. 2021. Enriching language models with visually-grounded word vectors and the Lancaster sensorimotor norms. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 148–157, Online. Association for Computational Linguistics.

[64] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

[65] Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *Proceedings of the 2019 Conference on Empirical Methods in*

Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1236–1242, Hong Kong, China. Association for Computational Linguistics.

[66] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237, Barcelona, Spain. Association for Computational Linguistics.

[67] Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of The Web Conference*, page 625–635.

[68] Vivek Kulkarni and William Yang Wang. 2018. Simple models for word formation in slang. 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 1424–1434, New Orleans, Louisiana. Association for Computational Linguistics.

[69] Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

[70] Kristian Kuznetsov, Eduard Tulchinskii, Laida Kushnareva, German Magai, Serguei Barannikov, Sergey Nikolenko, and Irina Piontkovskaya. 2024. Robust AI-generated text detection by restricted embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17036–17055, Miami, Florida, USA. Association for Computational Linguistics.

[71] JR Landis and GG Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159—174.

[72] Jey Han Lau, Paul Cook, Diana McCarthy, Spandana Gella, and Timothy Baldwin. 2014. Learning word sense distributions, detecting unattested senses and identifying novel senses using topic models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pages 259–270, Baltimore, Maryland. Association for Computational Linguistics.

[73] Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 591–601, Avignon, France. Association for Computational Linguistics.

[74] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proceedings of NIPS '14*, pages 2177–2185.

[75] Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

[76] Chen Li and Yang Liu. 2015. Joint pos tagging and text normalization for informal text. In *Proceedings of IJCAI*, pages 1263–1269.

[77] Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 165–174, New York, NY, USA. Association for Computing Machinery.

[78] Yucheng Li, Shun Wang, Chenghua Lin, and Frank Guerin. 2023. Metaphor detection via explicit basic meanings modelling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 91–100, Toronto, Canada. Association for Computational Linguistics.

[79] Yucheng Li, Shun Wang, Chenghua Lin, Frank Guerin, and Loic Barrault. 2023. FrameBERT: Conceptual metaphor detection with frame embedding learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1558–1563, Dubrovnik, Croatia. Association for Computational Linguistics.

[80] 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. *CoRR*, abs/1907.11692.

[81] Zhu Liu and Ying Liu. 2023. Ambiguity meets uncertainty: Investigating uncertainty estimation for word sense disambiguation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3963–3977, Toronto, Canada. Association for Computational Linguistics.

[82] Ismini Lourentzou, Kabir Manghnani, and ChengXiang Zhai. 2019. Adapting sequence to sequence models for text normalization in social media. *Proceedings of the ICWSM*, 13(01):335–345.

[83] Li Lucy and David Bamman. 2021. Characterizing English variation across social media communities with BERT. *Transactions of the Association for Computational Linguistics*, 9:538–556.

[84] Kikuo Maekawa, Makoto Yamazaki, Takehiko Maruyama, Masaya Yamaguchi, Hideki Ogura, Wakako Kashino, Toshinobu Ogiso, Hanae Koiso, and Yasuharu Den. 2010. Design, compilation, and preliminary analyses of balanced corpus of contemporary written Japanese. In *Proceedings of LREC '10*, pages 1483–1486.

[85] Kikuo Maekawa, Makoto Yamazaki, Takehiko Maruyama, Masaya Yamaguchi, Hideki Ogura, Wakako Kashino, Toshinobu Ogiso, Hanae Koiso, and Yasuharu Den. 2010. Design, compilation, and preliminary analyses of Balanced Corpus of Contemporary Written Japanese. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

[86] Walid Magdy, Yehia Elkhatib, Gareth Tyson, Sagar Joglekar, and Nishanth Sastry. 2022. Fake it till you make it: Fishing for catfishes. In *ASONAM '22: Proceedings of the 2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE Press.

[87] Matej Martinc, Petra Kralj Novak, and Senja Pollak. 2020. Leveraging contextual embeddings for detecting diachronic semantic shift. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4811–4819, Marseille, France. European Language Resources Association.

[88] Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of CoNLL '16*, pages 51–61.

[89] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

[90] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

[91] Bhaskar Mitra, Eric T. Nalisnick, Nick Craswell, and Rich Caruana. 2016. A dual embedding space model for document ranking. *arXiv: 1602.01137*.

[92] Syrielle Montariol, Matej Martinc, and Lidia Pivovarova. 2021. Scalable and interpretable semantic change detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4642–4652, Online. Association for Computational Linguistics.

[93] Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

[94] Ryo Nagata, Hiroya Takamura, Naoki Otani, and Yoshifumi Kawasaki. 2023. Variance matters: Detecting semantic differences without corpus/word alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15609–15622, Singapore. Association for Computational Linguistics.

[95] Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2).

[96] Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225, Uppsala, Sweden. Association for Computational Linguistics.

[97] Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

[98] 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 EMNLP '14*, pages 1059–1069.

[99] Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

[100] Ke Ni and William Yang Wang. 2017. Learning to explain non-standard English words and phrases. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 413–417, Taipei, Taiwan. Asian Federation of Natural Language Processing.

[101] Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: learning to define word embeddings in natural language. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3259–3266. AAAI Press.

[102] OpenAI. 2023. Introducing chatgpt.

[103] OpenAI. 2024. Gpt-4 technical report.

[104] Adam Pauls and Dan Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 959–968, Jeju Island, Korea. Association for Computational Linguistics.

[105] Zhengqi Pei, Zhewei Sun, and Yang Xu. 2019. Slang detection and identification. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 881–889, Hong Kong, China. Association for Computational Linguistics.

[106] Fuchun Peng, Dale Schuurmans, and Shaojun Wang. 2003. Language and task independent text categorization with simple language models. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 189–196.

[107] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[108] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

[109] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

[110] Ngoc-Quan Pham, Jan Niehues, and Alexander Waibel. 2018. Towards one-shot learning for rare-word translation with external experts. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 100–109, Melbourne, Australia. Association for Computational Linguistics.

[111] Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–217, Boulder, Colorado. Association for Computational Linguistics.

[112] Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of EACL '17*, pages 157–163.

[113] Xiao Pu, Nikolaos Pappas, James Henderson, and Andrei Popescu-Belis. 2018. Integrating weakly supervised word sense disambiguation into neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:635–649.

[114] 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. *J. Mach. Learn. Res.*, 21(1).

[115] Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of Workshop on NewChallenges for NLP Frameworks*, pages 45–50.

[116] Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1054, Melbourne, Australia. Association for Computational Linguistics.

[117] Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

[118] Itsumi Saito, Kugatsu Sadamitsu, Hisako Asano, and Yoshihiro Matsuo. 2014. Morphological analysis for Japanese noisy text based on character-level and word-level normalization. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1773–1782, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

[119] Itsumi Saito, Jun Suzuki, Kyosuke Nishida, Kugatsu Sadamitsu, Satoshi Kobashikawa, Ryo Masumura, Yuji Matsumoto, and Junji Tomita. 2017. Improving neural text normalization with data augmentation at character- and morphological levels. In *Proceedings of the Eighth International Joint Conference on*

Natural Language Processing (Volume 2: Short Papers), pages 257–262, Taipei, Taiwan. Asian Federation of Natural Language Processing.

[120] Aleksandr Sboev. 2016. The sources of new words and expressions in the Chinese Internet language and the ways by which they enter the Internet language. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation: Posters*, pages 355–361, Seoul, South Korea.

[121] Helmut Schmid. 1999. Improvements in part-of-speech tagging with an application to german. In *Natural Language Processing Using Very Large Corpora*, volume 11 of *Text, Speech and Language Processing*, pages 13–26.

[122] Vincent Segonne and Timothee Mickus. 2023. Definition modeling : To model definitions. generating definitions with little to no semantics. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 258–266, Nancy, France. Association for Computational Linguistics.

[123] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

[124] Philippa Shoemark, Farhana Ferdousi Liza, Dong Nguyen, Scott Hale, and Barbara McGillivray. 2019. Room to Glo: A systematic comparison of semantic change detection approaches with word embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

[125] Philippa Shoemark, Debnil Sur, Luke Shrimpton, Iain Murray, and Sharon Goldwater. 2017. Aye or naw, whit dae ye hink? Scottish independence and linguistic identity on social media. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1239–1248, Valencia, Spain. Association for Computational Linguistics.

[126] Pia Sommerauer and Antske Fokkens. 2019. Conceptual change and distributional semantic models: an exploratory study on pitfalls and possibilities. In *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, pages 223–233, Florence, Italy. Association for Computational Linguistics.

[127] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

[128] Ella Steen, Kathryn Yurechko, and Daniel Klug. 2023. You Can (Not) Say What You Want: Using Algospeak to Contest and Evade Algorithmic Content Moderation on TikTok. *Social Media + Society*, 9(3):20563051231194586.

[129] Ian Stewart and Jacob Eisenstein. 2018. Making "fetch" happen: The influence of social and linguistic context on nonstandard word growth and decline. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4360–4370, Brussels, Belgium. Association for Computational Linguistics.

[130] Zhewei Sun, Qian Hu, Rahul Gupta, Richard Zemel, and Yang Xu. 2024. Toward informal language processing: Knowledge of slang in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1683–1701, Mexico City, Mexico. Association for Computational Linguistics.

[131] Zhewei Sun, Richard Zemel, and Yang Xu. 2022. Semantically informed slang interpretation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5213–5231, Seattle, United States. Association for Computational Linguistics.

[132] Xiaohang Tang, Yi Zhou, Taichi Aida, Procheta Sen, and Danushka Bollegala. 2023. Can word sense distribution detect semantic changes of words? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3575–3590, Singapore. Association for Computational Linguistics.

[133] Sora Tarumoto, Koki Hatagaki, Rina Miyata, Tomoyuki Kajiwara, and Takashi Ninomiya. 2024. Evaluating chatgpt ' s ability to generate japanese. *Journal of Natural Language Processing*, 31(2):349–373.

[134] Arseny Tolmachev, Daisuke Kawahara, and Sadao Kurohashi. 2018. Juman++: A morphological analysis toolkit for scriptio continua. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 54–59, Brussels, Belgium. Association for Computational Linguistics.

[135] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

[136] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

[137] Haining Wang and Allen Riddell. 2022. CCTAA: A reproducible corpus for Chinese authorship attribution research. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5889–5893, Marseille, France. European Language Resources Association.

[138] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

[139] Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2985–3000, Dubrovnik, Croatia. Association for Computational Linguistics.

[140] Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artificial Intelligence*, 6(1):53–74.

[141] Yorick Wilks. 1978. Making preferences more active. *Artificial Intelligence*, 11(3):197–223.

[142] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

[143] Xiaobao Wu, Anh Tuan Luu, and Xinshuai Dong. 2022. Mitigating data sparsity for short text topic modeling by topic-semantic contrastive learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2748–2760, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[144] Hu Xu, Lei Shu, Philip Yu, and Bing Liu. 2020. Understanding pre-trained BERT for aspect-based sentiment analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 244–250, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[145] Muheng Yan, Yu-Ru Lin, Rebecca Hwa, Ali Mert Ertugrul, Meiqi Guo, and Wen-Ting Chung. 2020. Mimicprop: Learning to incorporate lexicon knowledge into distributed word representation for social media analysis. *Proceedings of ICWSM*, 14(1):738–749.

[146] Dongjie Yang, Zhuosheng Zhang, and Hai Zhao. 2023. Learning better masking for better language model pre-training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7255–7267, Toronto, Canada. Association for Computational Linguistics.

[147] Yi Yang and Jacob Eisenstein. 2017. Overcoming language variation in sentiment analysis with social attention. *Transactions of the Association for Computational Linguistics*, 5:295–307.

[148] Boliang Zhang, Hongzhao Huang, Xiaoman Pan, Sujian Li, Chin-Yew Lin, Heng Ji, Kevin Knight, Zhen Wen, Yizhou Sun, Jiawei Han, and Bulent Yener. 2015. Context-aware entity morph decoding. In *Proceedings of the 53rd Annual Meeting*

of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 586–595, Beijing, China. Association for Computational Linguistics.

[149] Yuhui Zhang, Chenghao Yang, Zhengping Zhou, and Zhiyuan Liu. 2020. Enhancing transformer with sememe knowledge. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 177–184, Online. Association for Computational Linguistics.

[150] Zhixue Zhao and Nikolaos Aletras. 2024. Comparing explanation faithfulness between multilingual and monolingual fine-tuned language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3226–3244, Mexico City, Mexico. Association for Computational Linguistics.

[151] Pei Zhou, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2024. Self-discover: Large language models self-compose reasoning structures. *arXiv preprint arXiv:2402.03620*.

[152] Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy. Association for Computational Linguistics.

Publication

Thesis-related

Journal Papers

- Tatsuya Aoki, Jey Han Lau, Hidetaka Kamigaito, Hiroya Takamura, Timothy Baldwin and Manabu Okumura, Discovering Unusual Word Usages with Masked Language Model via Pseudo-label Training. *Journal of Natural Language Processing*, Volume 32 Issue 1, 2025 (To be appeared).

- Tatsuya Aoki, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura, Detecting Non-standard Word Usages from Social Media. *Journal of Natural Language Processing*, Volume 26 Issue 2, pp. 381–406, 2019.

Others

Journal Papers

- Ying Zhang, Hidetaka Kamigaito, Tatsuya Aoki, Hiroya Takamura, and Manabu Okumura, Generic mechanism for reducing repetitions in encoder-decoder models. *Journal of Natural Language Processing*, Volume 30 Issue 2, pp. 401–431, 2023.

- Kasumi Aoki, Akira Miyazawa, Tatsuya Ishigaki, Tatsuya Aoki, Hiroshi Noji, Keiichi Goshima, Ichiro Kobayashi, Hiroya Takamura and Yusuke Miyao, Controlling contents in data-to-document generation with human-designed topic labels, *Journal of Computer Speech & Language*, Volume 66, page. 101154, 2021.

Conference Papers

- Riku Kawamura, Tatsuya Aoki, Hidetaka Kamigaito, Hiroya Takamura and Manabu Okumura, Neural Text Normalization Leveraging Similarities of Strings and Sounds, *In Proceedings of the International Conference on Computational Linguistics*, pp. 2126-2131, 2021.

- Tatsuya Aoki, Akira Miyazawa, Kasumi Aoki, Keiichi Goshima, Tatsuya Ishigaki, Ichiro Kobayashi, HiroyaTakamura and Yusuke Miyao, Generating Market Comments Referring to External Resources, *In Proceedings of the International Conference on Natural Language Generation*, pp. 135-139, 2018.