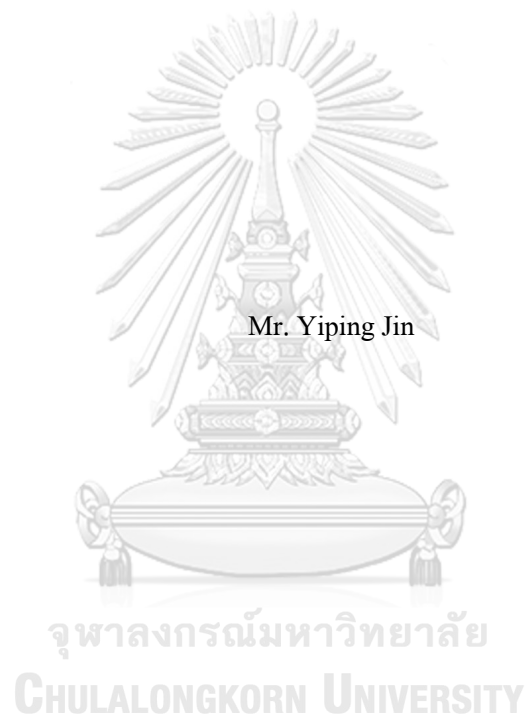


Natural Language Processing for Digital Advertising



Mr. Yiping Jin

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Computer Science and Information Technology

Department of Mathematics and Computer Science

FACULTY OF SCIENCE

Chulalongkorn University

Academic Year 2021

Copyright of Chulalongkorn University

การประมวลผลภาษาธรรมชาติสำหรับการโฆษณาดิจิทัล



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และวิทยาการ

คอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2564

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Thesis Title Natural Language Processing for Digital Advertising
By Mr. Yiping Jin
Field of Study Computer Science and Information Technology
Thesis Advisor Assistant Professor Dr. DITTAYA WANVARIE

Accepted by the FACULTY OF SCIENCE, Chulalongkorn University in Partial
Fulfillment of the Requirement for the Doctor of Philosophy

..... Dean of the FACULTY OF SCIENCE
(Professor Dr. POLKIT SANGVANICH)

DISSERTATION COMMITTEE

..... Chairman
(Dr. Choochart Haruechaiyasak)

..... Thesis Advisor
(Assistant Professor Dr. DITTAYA WANVARIE)

..... Examiner
(Associate Professor Dr. NAGUL COOHAROJANANONE)

..... Examiner
(Dr. Ekapol Chuangsuwanich)

..... Examiner
(Dr. NARUEMON PRATANWANICH)

ยี่ปิง จิน : การประมวลผลภาษาธรรมชาติสำหรับการโฆษณาดิจิทัล. (Natural Language Processing for Digital Advertising) อ.ที่ปรึกษาหลัก : ผศ. ดร.พิทยา หวานวารี

การโฆษณานั้นไม่ได้เป็นเพียงกิจกรรมการตลาดหรือการขาย แต่เป็นการสื่อสารสองทางรูปแบบหนึ่ง ในวิทยานิพนธ์นี้ ผู้วิจัยนำเสนอการประยุกต์งานการประมวลผลภาษาธรรมชาติ (natural language processing) 2 งาน ได้แก่ การเข้าใจภาษาธรรมชาติ (natural language understanding) และ การสังเคราะห์ภาษาธรรมชาติ (natural language generation) กับงานโฆษณาดิจิทัล เพื่อเพิ่มประสิทธิภาพในการโฆษณา

ผู้วิจัยประยุกต์ใช้การจำแนกข้อความแบบมีผู้สอนเล็กน้อยเพื่อสร้างตัวแบบจำแนกข้อความสำหรับการโฆษณาโดยอิงบริบทได้อย่างรวดเร็ว (Jin et al. 2022) วิธีนี้ต้องใช้การกำกับคำสำคัญเพียงเล็กน้อย แทนที่จะใช้คลังข้อความขนาดใหญ่ที่มีการกำกับชนิดของเอกสาร นอกจากนี้ วิธีนี้ยังสามารถนำไปใช้กับโดเมนใหม่ๆ ได้ง่ายอีกด้วย ผู้วิจัยยังประเมินผลตัวแบบซึ่งมีผู้สอนเล็กน้อยโดยใช้การประมาณค่าผิดพลาดแบบไม่มีผู้สอน และเลือกคำสำคัญแบบอัตโนมัติ (Jin et al. 2021a) การประมาณค่าผิดพลาดแบบไม่มีผู้สอนนั้นจำเป็น เนื่องจากเมื่อใช้วิธีการจำแนกข้อความแบบมีผู้สอนเล็กน้อยในสถานการณ์จริงจะไม่มีชุดข้อมูลที่มีการกำกับผลลัพธ์

ตัวแบบทรานส์ฟอร์มเมอร์ (Transformer) เป็นตัวแบบที่ดีที่สุดในการแปลงข้อความ เป็นข้อความ ผู้วิจัยใช้ตัวแบบทรานส์ฟอร์มเมอร์ในการสร้างคำโฆษณาที่เกี่ยวข้องและมีความหลากหลายจากคำอธิบายสั้นๆ ของบริษัท (Jin et al., In press) ผู้วิจัยป้องกันการรั่วข้อมูลที่ไม่สนับสนุนบริษัทจากการปิดช่องโหว่ในการฝึกสอน และสร้างคำโฆษณาที่หลากหลาย นำดึงดูด โดยใช้การฝึกสอนแบบมีเงื่อนไข

สาขาวิชา วิทยาการคอมพิวเตอร์และ เทคโนโลยีสารสนเทศ

ลายมือชื่อนิติ
.....

ปีการศึกษา 2564

ลายมือชื่อ อ.ที่ปรึกษาหลัก
.....

6173105023 : MAJOR COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

KEYWORD: Digital Advertising, Natural language processing, Natural language understanding, Natural language generation

Yiping Jin : Natural Language Processing for Digital Advertising. Advisor: Asst. Prof. Dr. DITTAYA WANVARIE

Advertising is not only a marketing or sales activity but a particular form of *two-way* communication. In this thesis, we propose to apply the two main subtasks of natural language processing (NLP), namely natural language understanding (NLU) and natural language generation (NLG), to digital advertising to enhance the effectiveness of advertising.

We apply weakly-supervised text classification to rapidly build text classifiers for contextual advertising (Jin et al. 2022). The method requires a handful of labeled keywords instead of a large corpus of labeled documents and can be easily transferred to new domains. We further evaluate the weakly-supervised models using unsupervised error estimation and perform automatic keyword selection (Jin et al., 2021a). Unsupervised error estimation is essential because no labeled development dataset is available in real-world problems where weakly-supervised text classification methods are applied.

Finally, we tap on a state-of-the-art sequence-to-sequence Transformer model to generate cohesive and diverse advertising slogans from a short company description (Jin et al., In press). We prevent the model from hallucinating unsupported information using entity masking and generate diverse and catchy slogans using conditional training.

Field of Study:	Computer Science and Information Technology	Student's Signature
Academic Year:	2021	Advisor's Signature

ACKNOWLEDGEMENTS

First and most importantly, I would like to thank my co-authors, Akshay Bhatia and Vishakha Kadam, who have contributed tremendously by implementing various challenging baselines. Without their help, the progress of my Ph.D. research would be drastically delayed.

I would also like to thank my advisor Asst. Prof. Dr. Dittaya Wanvarie for her constant guidance and support.

I appreciate my thesis committee's effort and constructive feedback, including Dr. Choochart Haruechaiyasak, Assoc. Prof. Dr. Nagul Cooharajanone, Assist. Prof. Dr. Ekapol Chuangsuwanich, and Dr. Naruemon Pratanwanich.

My heartfelt thanks also go to members of my research group in NUS who have guided me during my early research career, especially Assoc. Prof. Min-Yen Kan, Dr. Jun-Ping Ng, and Prof. Xiangnan He. I also benefitted greatly from personal correspondence with my long-time friend, Dr. Wenqiang Lei.

Last but not least, I would like to thank Chulalongkorn University and the graduate school for providing me this opportunity to study a Ph.D. program. Though I am leaving Thailand soon, I will for sure bring the beautiful memory with me as I open a new chapter of my life in a new continent.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Yiping Jin

TABLE OF CONTENTS

	Page
.....	iii
ABSTRACT (THAI)	iii
.....	iv
ABSTRACT (ENGLISH).....	iv
ACKNOWLEDGEMENTS.....	v
TABLE OF CONTENTS	vi
LIST OF TABLES.....	ix
LIST OF FIGURES	xi
1. Introduction.....	1
1.1 Digital Advertising.....	1
1.2 Contextual Advertising.....	2
1.3 Slogan Generation.....	3
1.4 Dissertation Preview.....	4
2. Contextual Advertising with Weak-Supervision	5
2.1 Related Work	6
2.1.1 Weakly-supervised text classification.....	6
2.1.2 Text classification with noisy labels	7
2.1.3 Domain adaptation.....	8
2.2 Proposed Method	9
2.2.1 Mining keywords from noisy corpus	10
2.2.2 Training weakly-supervised text classifier	12

2.3 Experiments.....	14
2.3.1 Datasets	14
2.3.2 Experimental setup and baselines	16
2.3.2.1 Parameter setting	16
2.3.2.2 Baselines	16
2.3.2.3 Performance metrics.....	17
2.3.3 Results and discussion.....	18
2.3.3.1 Mined keywords from labeled corpus	18
2.3.3.2 Text classification performance	18
2.3.3.3 Impact of keyword selection strategy	24
2.3.3.4 Domain adaptation performance.....	25
2.4 Conclusions	26
3. Evaluating Weakly-Supervised Classifiers with Bayesian Error Estimation.....	27
3.1 Related Work	28
3.1.1 Weakly-supervised text classification.....	28
3.1.2 Unsupervised error estimation.....	29
3.1.3 Keyword mining	29
3.2 Proposed Method	30
3.2.1 Mining candidate keywords from a single seed word	30
3.2.2 Training interim classifiers.....	30
3.2.3 Evaluating seed words with Bayesian error estimation	31
3.3 Experiments.....	33
3.3.1 Experimental setup	33
3.3.2 Classification performance.....	34

3.3.3 Case study	36
3.4 Conclusions	36
4. Generating Slogans with Seq2seq Transformers	38
4.1 Related Work	38
4.1.1 Slogan generation	38
4.1.2 Sequence-to-sequence models	40
4.2 Datasets	42
4.3 Proposed Method	45
4.3.1 Model	45
4.3.2 Generating truthful slogans with masking	46
4.3.3 Generating diverse slogans with syntactic control	48
4.4 Experiments.....	50
4.4.1 Quantitative evaluation.....	50
4.4.2 Truthful evaluation	53
4.4.3 Diversity evaluation.....	54
4.4.4 Human evaluation.....	56
4.5 Conclusions	57
5. Conclusions	58
REFERENCES	59
VITA	70

LIST OF TABLES

Table 1: The number of labels assigned to documents in the RTB dataset.	14
Table 2: Dataset statistics.	15
Table 3: Mined keywords using pmi-freq.	19
Table 4: Each Model’s performance on three evaluation datasets. The best results for each metric are in boldface.	20
Table 5: Five most frequent categories STM predicted for documents in the pets category.	22
Table 6: Sample documents with label ‘Pets’ that STM classifies as “Family and parenting.”	22
Table 7: STM’s performance using different sets of keywords.	25
Table 8: Domain adaptation performance using unlabeled in-domain data.	26
Table 9: Initial seed words for each classification task.	34
Table 10: Accuracy on topic classification tasks. cate , ours , gold represent using the category name, OptimSeed seed words, and human-curated seed words reported in previous work. We highlight the best-performing keyword set for each model.	35
Table 11: Accuracy on sentiment classification tasks. We highlight the best-performing seed words in bold for each model-task combination.	35
Table 12: Average accuracy scores over six classification tasks. We highlight the best performing seed words for each model in bold. * denotes statistical significance using double-sided paired T-test with a p-value of 0.05 w.r.t. the same model using the “cate” seed words.	36
Table 13: Seed words for "Economy" at various steps within the OptimSeed framework.	36
Table 14: The percentage of descriptions and slogans containing each type of entity. "Slog-Desc" refers to the percentage of entities only occur in the slogan.	44
Table 15: Sample (description, slogan) pairs from the validation set. We highlight the exact match words in bold.	45
Table 16: Example description before and after company name delexicalization.	47
Table 17: Applying entity masking to an example description and slogan pair.	48

Table 18: The top 10 slogan POS tag sequences in the training data with example.	49
Table 19: Full list of syntactic control codes.	49
Table 20: The ROUGE -1/-2/-L F1 scores of various models on the validation and test datasets.	51
Table 21: Sample generated slogans from different models. "Gold" is the original slogan. The DistilBART model uses both company name delexicalization and entity masking.	52
Table 22: The automatic truthfulness evaluation scores of the baseline DistilBART model and our proposed method. The p-value of a double-sided paired t-test is presented in brackets.	53
Table 23: Different method's syntactic control accuracy and word diversity. We highlight the best scores in bold. All models do not use delexicalization or entity masking.	54
Table 24: Pair-wise human evaluation result of each control code compared with the nucleus sampling baseline. We calculate the p-value using double-sided Wilcoxon signed-rank test. "Better" indicates that our method generates better slogans than the baseline and vice versa for "worse"	55
Table 25: Randomly sampled slogans generated with different control codes.	56
Table 26: Human evaluation on coherence, well-formedness, and catchiness. We highlight the best score for each aspect in bold (excluding the "coherent" aspect for the first sentence baseline because it is coherent by definition). ** indicates statistical significance when compared with our method using a two-sided paired t-test using p-value=0.005.	57

LIST OF FIGURES

Figure 1: Real-Time Bidding Process. Adapted from Wang et al. 2017.....	2
Figure 2: Screenshot of The New York Times' Homepage	5
Figure 3: Number of correct keywords generated by each algorithm with different levels of label noise.	12
Figure 4: The seed-word guided topic model's workflow.....	13
Figure 5: STM's prediction confusion matrix on news-crawl test set. We removed the diagonal entries (correct predictions) to surface misclassifications.	21
Figure 6: LIME explanations on a sample where both models predicted the correct label.	23
Figure 7: LIME explanations on a sample document where STM predicted correctly, but ULMFiT predicted wrongly.	24
Figure 8: OptimSeed, a framework to select seed words for weakly-supervised text classification using unsupervised error estimation.....	28
Figure 9: Graphical model for error estimation.	33
Figure 10: Distribution of the number of (subword) tokens. Left: description. Right: slogan.	43
Figure 11: Distribution of the number of companies in each industry (in log-10 scale). X-axis is the number of companies belonging to an industry in log-10 scale. Y-axis is the number of industries in each bucket.....	43
Figure 12: The first-k words baseline's ROUGE scores by varying k.....	51

1. Introduction

Traditionally, brands who wish to advertise for their product or service had to contact publishers (e.g., newspapers, owners of billboards, television channels) and sign a deal to display their ads in their media. The idea of personalized and contextual advertising is not unique to the internet age (Langheinrich et al., 1999; Burton and Lichtenstein, 1988; Yi 1991). Advertisers may infer the audience's demographic, economic, or social background from the nature of the publisher. E.g., the target audience of Wall Street Journal is affluent and influential readers. They may also deliver contextually relevant ads by specifying in the contract which section they want to display their ads. E.g., it is intuitive to show a Nike ad in the "Sports" section of a newspaper.

However, digital advertising opened new possibilities to deliver personalized and contextualized advertisements. With advanced artificial intelligence technologies, we can now deliver the **right message** to the **right audience** in the **right context**. In [Section 1.1](#), we brief the eco-system of digital advertising, which enables real-time auction of advertising opportunities without any direct contract between advertisers and publishers. [Section 1.2](#) introduces contextual advertising, a key component to understand the users' browsing context. We describe the automatic slogan generation task in [Section 1.3](#). Finally, we overview the dissertation in [Section 1.4](#).

1.1 Digital Advertising

Traditionally, advertisers had to sign a deal with publishers directly to advertise on their media. It is not only time-consuming but also costly. Each publisher imposes a minimum ad spend requirement, usually at least thousands of dollars per month. If an advertiser wishes to advertise across different publishers, the monthly advertising cost will easily exceed tens of thousands of dollars.

Real-time bidding (RTB) solves this problem by providing a common marketplace for online advertising (Wang et al., 2017). Each ad opportunity (impression) is traded in an open auction without advertisers contacting the publishers directly. Figure 1 shows the real-time bidding process. Each time a user accesses a page, the site will send an ad request to the ad exchange if ad slots are available. The ad exchange will then send a bid request to selected advertisers. The bid request contains the URL and some other information, such as user location and time. The

advertisers optionally query an in-house or third-party data management platform to obtain the users' attributes such as age and interest. They then decide whether to submit a bid for the ad impression. After receiving bids from all advertisers, the ad exchange selects the advertiser with the highest bid and notifies them that they won the auction. Finally, the ad exchange returns the ad to the page to display to the user.

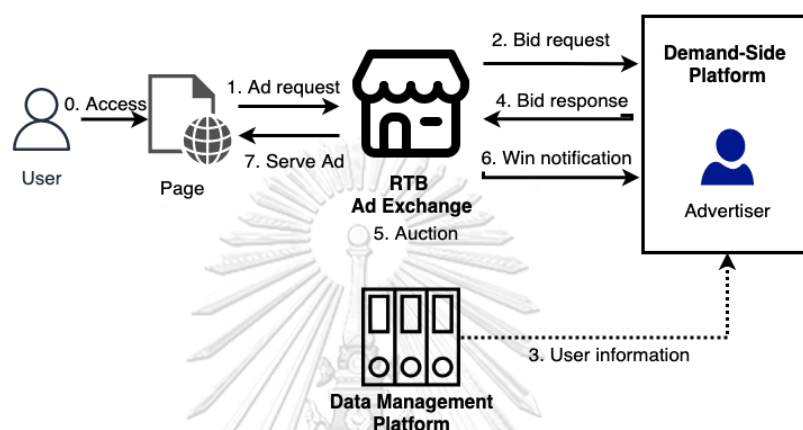


Figure 1: Real-Time Bidding Process. Adapted from Wang et al. 2017.

Real-time bidding is highly time-sensitive and large-scale. The entire process of real-time bidding takes place within 100 milliseconds, which is the page loading time. In addition, advertisers often process millions of bid requests per second. These characteristics make it an ideal playground for large-scale machine learning models to optimize the performance of ad campaigns. Understanding the user and context helps advertisers estimate the value of the ad impression and come up with the optimal bidding strategy.

1.2 Contextual Advertising

Contextual advertising (or contextual targeting) is an essential advertising technology to display advertisements on web pages about similar content (Jin et al., 2017, Jin et al., 2022). The relevance between the advertisements and the web page makes the ad less intrusive, and users are more likely to interact with the ad because they are in the right “context” or mindset.

Contextual advertising is usually performed by classifying web pages into a list of relevant categories, either from a predefined taxonomy or custom categories. For example, the Interactive

Advertising Bureau (IAB) curated a categorization taxonomy for online advertising¹ commonly used in the industry. The taxonomy consists of 23 tier-1 categories and 355 tier-2 categories. Additionally, there are usually thousands of active custom categories in an advertising platform because advertisers often want to customize the content they wish to target. The large number of categories and the dynamic nature makes traditional supervised text classification techniques unsuitable because they require a sizeable labeled dataset, which is time-consuming and expensive to obtain. Therefore, we focus on applying weakly-supervised text classification to contextual advertising without requiring any manually labeled document.

1.3 Slogan Generation

Advertisers use slogans to catch the viewers' attention and encourage them to perform the desired action, such as purchasing the item or interacting with the online ad. Early marketing research demonstrated that good slogans are **concise** (Lucas, 1934) and **creative** (White, 1972; Phillips and McQuarrie, 2009; Mieder and Mieder, 1977).

Until very recently, slogan writing remained the task of highly specialized human copywriters. Slogan writing is a particularly challenging and time-consuming task. Copywriters need to understand the unique selling points and apply creativity "within strict parameters." The customers should easily associate the slogans with the product, i.e., the slogans should be **coherent** with the product. On the other hand, even a good slogan's effectiveness decreases over time due to ad fatigue (Abrams and Vee, 2007). Therefore, copywriters have to compose multiple slogans for the same ad campaign and change them regularly.

Previous work in automatic slogan generation focused on mining and utilizing slogan skeletons (Özbal et al., 2013; Tomašić et al., 2014; Gatti et al., 2015; Alnajjar and Toivonen, 2021), such as "Think Big, Think [PRODUCT]." However, the slogan skeletons from a random company in the slogan database are not likely coherent with a particular product.

Some recent work applied sequence-to-sequence (seq2seq) models to generate new slogans from scratch (Misawa et al., 2020; Hughes et al., 2019). However, they did not utilize the recently proposed Transformers architecture (Vaswani et al., 2017), which dominates most natural

¹ <https://www.iab.com/guidelines/iab-quality-assurance-/guidelines-qag-taxonomy/>.

language generation leaderboards. In this dissertation, we apply a transformers encoder-decoder model to slogan generation. We also use techniques to improve the truthfulness and diversity of generated slogans.

1.4 Dissertation Preview

This dissertation studies the application of natural language understanding and generation to digital advertising. The goal is to produce more effective contextual advertising and slogan generation systems to optimize campaign performance and automate the campaign creation process.

We organize the rest of this dissertation as follows: [Chapter 2](#) introduces a weakly-supervised text classification model that requires only a handful of labeled keywords. The keywords can be either automatically mined from a noisily-labeled out-of-domain corpus or curated by domain experts.

[Chapter 3](#) presents a novel unsupervised evaluation method for weakly-supervised models without requiring any labeled development dataset. We further utilize the unsupervised evaluation result to perform automatic keyword selection.

Finally, [Chapter 4](#) introduces a sequence-to-sequence Transformer model to **generate** cohesive and diverse advertising slogans from a short company description. We prevent the model from hallucinating unsupported information using entity masking and generate diverse and catchy slogans using conditional training.

2. Contextual Advertising with Weak-Supervision

As motivated in the introduction section, contextual advertiser faces the challenge of an enormous number of categories. In addition, the web contains heterogeneous content. Newswire sites often categorize the articles. For example, Figure 2 shows a screenshot of the New York Times’ homepage. Each article is categorized into one of the highlighted categories. We can utilize the news categories by mapping them to the categories of our target IAB content taxonomy. It allows us to crawl a large labeled dataset from newswire sites without any manual labeling.



Figure 2: Screenshot of The New York Times’ Homepage

However, this gives rise to two additional problems. Firstly, some of the documents are *miscategorized*. E.g., a gossip article about an athlete might be assigned to the “Sports” category, but its content is most relevant to “Arts & Entertainment.” The more subtle problem is that the news articles are homogenous in style and length and do not resemble general web content such as forum posts and home pages. Thus, the performance might deteriorate if we train a text classifier only on news articles and use it to classify *out-of-domain* data (as we shall demonstrate in Section 2.3).

This chapter studies how to apply weakly-supervised text classification methods to mitigate the noisy label and out-of-domain problem. Specifically, we mine representative keywords from the automatically labeled news corpus and apply weakly-supervised learning on unlabeled in-domain documents. We make the following three main contributions:

- We mine keywords from a noisy corpus using a robust statistical method and use the keywords as a bridge between domains.
- The proposed method significantly outperformed strong supervised learning baselines without using any labeled document.

- We analyze the working of the classifiers to explain why the proposed method yields superior performance, which supports future theoretical and empirical studies.

2.1 Related Work

2.1.1 Weakly-supervised text classification

Weakly-supervised text classification (Chang et al., 2008) induces classifiers using *labeled keywords* and *an unlabelled corpus*. There are various approaches for weakly-supervised classification, such as hand-crafted rules, constraint optimization, injecting the keywords as priors to the model, and semantic representation of the documents and the labels.

Chang et al. (2008) used *Explicit Semantic Analysis (ESA)* (Gabrilovich et al., 2007) to represent both the category and the documents in a shared semantic space, whose dimensions are Wikipedia concepts. During inference, they calculate the cosine similarity between the document and category representation and predict the nearest category to the document. They also applied weakly-supervised classification to domain adaptation. However, they only considered the binary classification between two categories, “baseball” and “hockey,” and their source and target dataset (20NG and Yahoo! Answers dataset) are both manually labeled and do not contain noisy labels.

Druck et al. (2008) proposed generalized expectation (GE) criteria, which trains classifiers by performing constraint optimization over the distribution of labeled keywords among documents predicted into each category. GE can extend to different tasks, including text categorization (Druck et al., 2008) and language identification in mixed-language documents (King and Abney, 2013). Similarly, Charoenphakdee et al. (2019) introduced a theoretically proved risk minimization framework that directly optimizes the area under the receiver operating characteristic curve (AUC) of a weakly-supervised classification model.

Settles (2011) and Li and Yang (2018) both applied multinomial naïve Bayes (MNB) to weakly-supervised classification. Settles (2011) increased the Dirichlet prior for labeled keywords. His method involves three steps: estimate the initial parameters using only the priors, apply the initial classifier on unlabelled documents and re-estimate the model parameters using labeled and pseudo-labeled documents. In contrast, Li and Yang (2018) used the labeled keywords to directly pseudo-label documents. They then applied standard semi-supervised learning using the EM algorithm.

Li et al. (2016) proposed Seed-Guided Topic Model (STM). STM models two types of topics: *category-topics* and *general-topics*. Category-topics contain content words that are specific to a category. General-topics involve frequent words which do not indicate a category. For example, the presence of the keyword “mammogram” almost certainly suggests that the document is related to cancer. However, although keywords like “breast” and “prostate” frequently occur in documents about “cancer,” they alone are not sufficient clues to label the document as “cancer.” STM is trained in two steps: first, initializing the category word probability and the document category distribution based on the co-occurrence with labeled seed words. Then, they apply joint Gibbs sampling to infer all the hidden parameters. STM significantly outperformed various baselines, including GE and a naïve Bayes model similar to Settles (2011).

Meng et al. (2018) proposed WeSTClass, a novel weakly-supervised neural text classification framework. Firstly, it generates *pseudo documents* using a generative mixture model, which repeatedly generates several terms from the background and class-specific distributions. The pseudo documents are used to *pre-train* a neural model. To adapt to real-world input documents, it performs self-training on unlabelled real documents and automatically adds the most confident predictions to the training set. The method outperformed baselines such as IR with TF-IDF, Chang et al. (2008), and CNN trained on pseudo-labeled documents.

In practice, weakly-supervised classification models can often yield performance close to a fully-supervised model. Because weakly-supervised classification does not require any labeled document, we can easily retrain the model if we notice a domain drift. However, a supervised model remains static once the data labeling is complete. Any further data labeling will usually involve a substantial cost.

2.1.2 Text classification with noisy labels

Label noise is prevalent in real-world scenarios, and it can have a significant negative impact on classifier’s accuracies (Frénay and Verleysen, 2014). It is especially severe for recent over-parameterized deep learning models with hundreds of millions of parameters. Such models can easily overfit any anomaly and cause low generalization performance. There are mainly three techniques to deal with label noise in text classification: label *noise-robust* models, *data cleansing* methods, and *noise-tolerant* methods.

Label *noise-robust* models is the simplest method. It does not aim to remove or take into consideration the label noise. Dietterich (2000) demonstrated that some models are robust to label noise by design, such as ensembles using bagging. In contrast, although the support vector machine (SVM) was established as a strong baseline for many classification tasks, it is sensitive to label noise because it relies on a few support vectors near the decision boundary (Nettleton et al., 2010). In general, label noise-robust models are effective when the wrongly labeled instances are relatively rare (<10%).

Data cleansing methods aim to either remove wrongly labeled data or correct their labels before training the classifier. One crucial advantage of data cleansing methods is that it is a separate *preprocessing step* and can be combined with any subsequent classification algorithm. Standard procedures of data cleansing involve anomaly detection (Sun et al., 2007) or prediction-based filtering using k-fold cross-validation (Gamberger et al., 1999) or voting (Brodley et al., 1996).

The final and most complex method dealing with label noise is *noise-tolerant* models. They try to model the label noise explicitly within the classifier, often using Bayesian priors (Swartz et al., 2004; Gerlach and Stamey, 2007). Similarly, Breve et al. (2010) applied a semi-supervised graph-based algorithm to perform label propagation among similar examples to correct the labels of wrongly labeled examples.

2.1.3 Domain adaptation

Classifiers are often trained once and applied in production systems for an extended period. As a result, data drift can still happen even if the researchers carefully consider the application and closely resemble the real-world data distribution in their training dataset. Domain adaptation aims to mitigate the negative impact of data drifts by “adapting” model M to M' , which is more robust to incoming data dissimilar to the original training instances. There are two distinct scenarios for domain adaptation depending on whether in-domain labeled data are available.

Transfer learning (Pan and Yang, 2010) is the standard method when in-domain labeled data are available. It was popularized by the ImageNet challenge (Krizhevsky et al., 2012) from the image processing community. In transfer learning, we first *pre-train* a model on a *large* and *general-purpose* dataset, often consisting of millions of instances. We then *fine-tune* the model to

the target classification task or domain with much less data. Fine-tuning is usually performed by replacing only the final classification layer(s) and maintaining most pre-trained models' parameters unchanged. Since 2018, transfer learning has also dominated the NLP field and yielded new state-of-the-art results across the board (Howard and Ruder, 2018; Peters et al., 2018). Analogous to the ImageNet models, they pre-trained a language model on a large text corpus using the next-word-prediction *unsupervised* learning objective. The language model observed word and phrase usage in abundant contexts, making them an ideal starting point for downstream language understanding and generation tasks. Consequently, transfer learning drastically reduces the amount of labeled data required to train classifiers. For example, Howard and Ruder (2018) showed that they could achieve the same performance as the model trained from scratch using only 1/100 of the data (100 labeled examples).

When no in-domain labeled data are available, we need to either build models that are robust to different domains (analogous to “noise-robust” methods mentioned in the previous section) or apply unsupervised learning. Sachan et al. (2018) quantified various models' reliance on the presence of keywords by carefully constructing training and testing datasets without key lexicon overlap. Modern deep learning models bragged about utilizing contextualized information instead of simply relying on keywords (Peters et al., 2018). However, Sachan et al. (2018) demonstrated that such models still rely heavily on keywords since the performance dropped on average by 10-20% on the new test set. To this end, they proposed keyword anonymization and adaptive word dropout to regularize the model and make it less reliant on the presence of keywords.

Mudinas et al. (2018) proposed an unsupervised learning method to bootstrap domain-specific sentiment classifiers. They noticed that the sentiment words with opposite polarity form distinct clusters. Therefore, they trained a simple linear classifier to separate positive and negative sentiment words, resulting in a sentiment lexicon. Subsequently, they used the lexicon to pseudo-label an unlabeled dataset, which is used to train their final classifier.

2.2 Proposed Method

We propose a two-step approach to address noisy out-of-domain classification. First, we mine keywords from a noisily labeled corpus (Section 2.2.1). Second, we apply weakly-supervised text classification to induce classifiers from the keywords and an unlabeled in-domain corpus (Section 2.2.2).

2.2.1 Mining keywords from noisy corpus

Weakly-supervised learning’s effectiveness depends substantially on the quality of seed keywords (Li et al., 2018; Jin et al., 2022). Previous work used either manually curated keywords (Druck et al., 2008; Settles, 2011; Meng et al., 2018) or limited to the category name or category description (Chang et al., 2008; Li et al., 2016; Li et al., 2018). The former depends on domain knowledge, and we cannot conclude whether the reported result is due to the model’s superiority or the careful choice of seed keywords. The latter may lack expressive power since some category names are ambiguous. Instead, we propose to automatically mine seed keywords from a noisy corpus for weakly-supervised models in this work. We define the task formally as follows.

We have a corpus $(\mathbf{D}_1, \dots, \mathbf{D}_C)$, where $\mathbf{D}_c = (d_{c,1}, \dots, d_{c,k})$ is the set of labeled documents for category c . Each document $d_{c,i}$ consists of a list of words (w_1, \dots, w_j) . Our goal is to generate a set of representative keywords k_1, \dots, k_n from the vocabulary $V = [w_1, \dots, w_N]$ for each category. This formulation is related to the strength of association as measured in information theory. We first calculate pointwise mutual information (*pmi*) between keyword w and category c . $pmi(w;c)$ is defined as follows:

$$pmi(w;c) \equiv \log \frac{p(w,c)}{p(w)p(c)} = \log \frac{df(w,c) \sum_{c \in C} df(c)}{df(w)df(c)} \quad (1)$$

Where $df(w,c)$ is the count of documents from category c containing the word w ; $df(w)$ is the count of documents containing the word w ; $df(c)$ is the count of documents from category c . Our initial experiments indicate that pointwise mutual information favors rare words that cooccurred with a category by random chance. Therefore, we modified the *pmi* score by multiplying it with the logarithmic term-frequency of word w and setting a threshold to block rare words. The new metric *pmi-freq* is defined as follows.

$$pmi - freq(w;c) \equiv \begin{cases} \log df(w) \cdot pmi(w;c), & \text{if } df(w) \geq 5 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The *pmi-freq* score is calculated independently for each category. However, it is also ideal to ensure that the keywords mined for each category have little overlap. To this end, we apply maximal marginal relevance (*mmr*) (Carbonell and Goldstein, 1998) to penalize keywords that rank high in multiple categories.

$$mmr(w_i; c_m) \equiv \arg \max_{w_i} [\lambda \cdot Sim(w_i, c_m) - (1 - \lambda) \max_{m \neq n} Sim(w_i, c_n)] \quad (3)$$

The first term measures the relatedness between word w_i and category c_m . The second term indicates the maximum relatedness between w_i and another category c_n . Intuitively, for a word to be ranked high w.r.t. category c , it must have a high score for the first term and a low score for the second term. The parameter λ controls the weights of the two terms. We use a default $\lambda=0.5$ and *pmi-freq* as the similarity measure for both terms.

We want to study how robust each keyword mining algorithm is to the label noise since we intend to apply them to noisily labeled data. We generate synthesized label noise with different label corruption rates and conduct an intrinsic evaluation of mined keywords' quality. Specifically, we experiment with the 20 newsgroups dataset (Lang, 1995) and vary the label corruption rate from 0% up to 70%. We randomly picked two categories (automotive, baseball) to manually count the number of correct keywords among the top 10 keywords mined by each algorithm.

Druck et al. (2008) proposed to mine keywords using mutual information (*mi*), which is defined as follows:

$$mi(w; C) = \sum_{c \in C} p(w, c) \cdot pmi(w, c) \quad (4)$$

A key difference between *mi* and *pmi* is that *mi* is not specific to a particular category since it sums the weighted *pmi* scores for all categories. As a result, it rewards keywords that are frequent in multiple categories. Druck et al. (2008) assigned each keyword to the category where it occurs most often.

We compare the aforementioned keyword mining algorithms and a naïve baseline *freq*, selecting the most frequent word appearing in each category (after removing the stop words). Figure 3 presents the number of correct keywords each algorithm generates at a different level of label noise rate. We can observe that *pmi-freq* and *mmr* almost always outperform *pmi*, especially when the label noise rate is high. On the other hand, the *mi* and *freq* baseline both performed poorly. Their performance also fluctuates much more than our proposed method. This intrinsic evaluation validates the effectiveness of our proposed approach to mine high-quality keywords, even when up to 50% of the document labels are corrupted. As *pmi-freq* and *mmr* perform on par with each other, we use *pmi-freq* subsequently due to its simplicity.

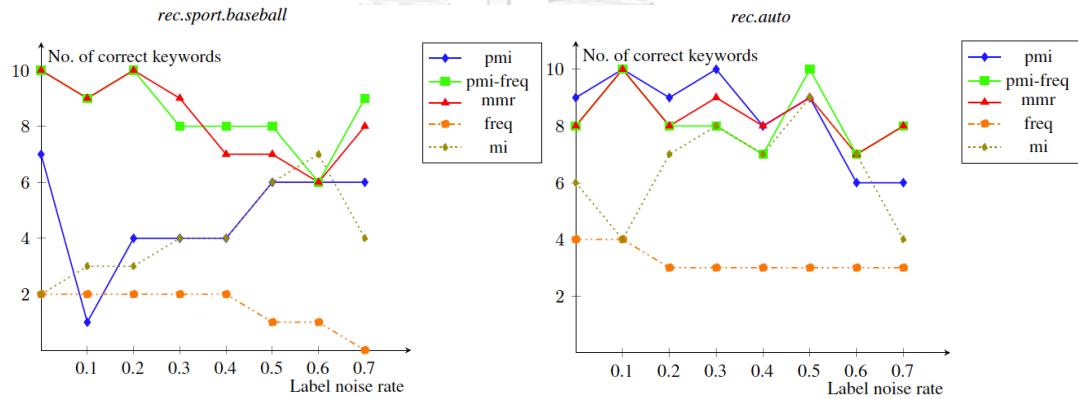


Figure 3: Number of correct keywords generated by each algorithm with different levels of label noise.

2.2.2 Training weakly-supervised text classifier

We use STM (Li et al., 2016) as the learning algorithm, whose workflow is depicted in Figure 4. STM takes labeled keywords (seed words) and unlabeled documents as input. The model estimates the initial document category distribution by counting labeled keywords. It also calculates the probability of each unlabeled word belonging to each category using their co-occurrence with the labeled keywords.

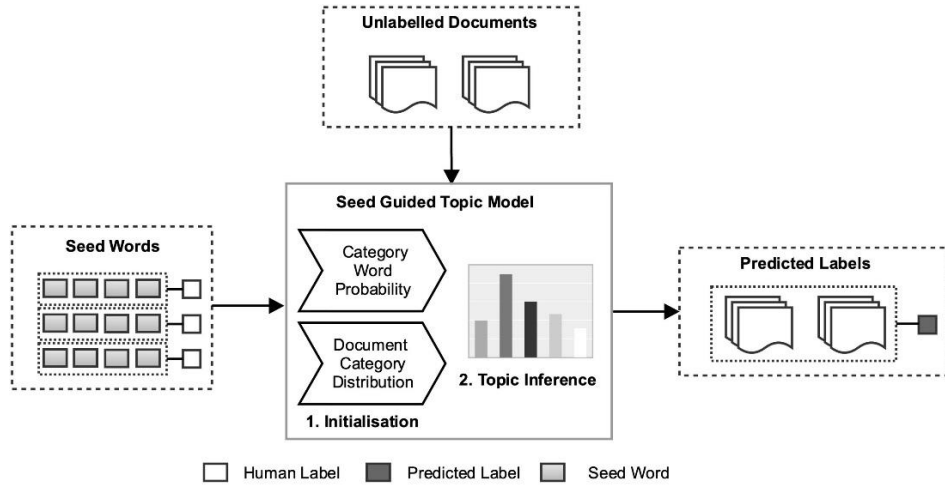


Figure 4: The seed-word guided topic model's workflow.

STM captures two types of topics: the general topic and the category topic. General topics capture global semantic information and are common to all documents. On the other hand, category topics are specific to a single category and contain category-indicating keywords. To make the distinction, STM introduces a latent binary variable $x_{d,i}$, which indicates whether the word $w_{d,i}$ is generated from document d 's category topic c_d or one of the general topics. The generative process of the Gibbs Sampling inference is detailed in Algorithm 1.

1. for each category $c \in \{1 \dots C\}$,
 - a) draw a general-topic distribution $\varphi \sim \text{Dirichlet}(\alpha_0)$ and
 - b) draw a category word distribution $\vartheta \sim \text{Dirichlet}(\beta_0)$;
 2. for each general topic $t \in \{1 \dots T\}$,
 - (a) draw a word distribution for the general topic $\phi_t \sim \text{Dirichlet}(\beta_1)$;
 3. For each document $d \in \{1 \dots D\}$,
 - (a) generate an initial category distribution η_d ;
 - (b) draw category $c_d \sim \text{Multinomial}(\eta_d)$;
 - (c) draw a general-topic distribution $\theta_d \sim \text{Dirichlet}(\alpha_1 \cdot \varphi_{c_d})$;
 - (d) for each word $i \in \{1 \dots |d|\}$,
 - i. draw $x_{d,i} \sim \text{Bernoulli}(\delta_{w_{d,i}, c_d})$;
 - ii. if $x_{d,i} = 0$: draw word $w_{d,i} \sim \text{Multinomial}(\vartheta_{c_d})$;
- if $x_{d,i} = 1$:
- A. draw general-topic assignment $z_{d,i} \sim \text{Multinomial}(\theta_d)$;
 - B. draw word $w_{d,i} \sim \phi_{z_{d,i}}$.

Algorithm 1: Gibbs Sampling generative process for STM.

Two factors lead to STM's success as a weakly-supervised classification algorithm. Firstly, the model utilized the co-occurrence between unlabeled and labeled keywords to initialize the category word probability. Although the initial probability is imperfect, it is much better than an

uninformative uniform probability, thus providing a better starting point for the subsequent inference algorithm. Secondly, the model’s separation of general and category topics allows it to focus on reliable category-indicating keywords while skimming through the rest of the document.

2.3 Experiments

2.3.1 Datasets

We conduct experiments on three datasets. The first one is a legacy large labeled dataset crawled from newswire websites (*news-crawl* dataset). We manually map the news categories of each website to IAB tier-1 categories, and we did not manually verify each document’s label correctness because it is costly and time-consuming. We crawled another evaluation dataset recently with approximately 100 documents per category using a similar method (*news-crawl-v2* dataset). The two datasets are collected during different periods: *news-crawl* dataset was collected before April 2015, while *news-crawl-v2* dataset was collected in May 2019. Besides, they come from different sets of websites. These differences allow us to study how models perform on slightly different domains.

The last dataset is a small manually curated evaluation dataset (*RTB* dataset) coming from in-domain real-time bidding (RTB) traffic. RTB contains heterogeneous content, such as blogs, forums, which are substantially different from news pages. All three datasets have the same 22 categories. We release the evaluation datasets publicly for future work to compare with our method².

While labeling the *RTB* dataset, we realized that some documents could belong to more than one category. Therefore, we allow annotators to assign multiple categories to a document. Table 1 presents the number of documents with different numbers of assigned labels. We can observe that the vast majority of the documents belong to one or two categories.

No. of assigned labels	No. of documents
1	892
2	516
3	84
4	9

Table 1: The number of labels assigned to documents in the *RTB* dataset.

² <https://github.com/YipingNUS/nle-supplementary-dataset>

We overview the number of documents in the three datasets in Table 2. The average document length for *news-crawl*, *news-crawl-v2*, and *RTB* datasets are 503, 1,470, and 350 words separately, suggesting a potential discrepancy between the training data and the data the model is applied to.

Category	News-crawl	News-crawl-v2	RTB
Business	44,343	100	50
Society	25,460	89	71
Technology and computing	16,466	100	178
Health and fitness	16,171	100	132
Law, government, and politics	14,374	97	44
Science	11,863	100	96
Sports	11,055	100	92
Art and entertainment	10,746	100	207
Education	8,321	100	80
Personal finance	5,693	80	56
Automotive	5,522	91	109
Food and drinks	4,408	100	173
Family and parenting	4,204	118	44
Style and fashion	4,191	100	62
Travel	3,995	100	135
Hobby and interest	3,710	100	117
Pets	3,246	100	22
Religion and spirituality	2,936	95	57
Home and garden	2,427	100	66
Real estate	2,056	100	86
Careers	1,685	65	49
Shopping	1,611	92	152
Total	204,483	2,127	1,501

Table 2: Dataset statistics.

2.3.2 Experimental setup and baselines

We randomly split *news-crawl* dataset into 90%/10% training and testing datasets. Then, we mine the keywords from the *labeled* training dataset. Finally, the testing split of *news-crawl* dataset and the other two datasets are used for evaluation only.

2.3.2.1 Parameter setting

We mine the top 15 single-word keywords for each category as ranked by *pmi-freq*. We lowercase all documents and exclude keywords shorter than four characters, contain digits, or are in a stopword dictionary. We train STM using the remaining keywords and the training dataset (with labels removed). We follow the parameters in Li et al. (2016) and perform inference for five epochs.

2.3.2.2 Baselines

We compare with various fully-supervised and weakly-supervised baselines to validate the effectiveness of our proposed method.

Fully-Supervised learning baselines:

- **Multinomial naïve Bayes (MNB):** a competitive baseline despite its simplicity (Wang and Manning, 2012). We train a supervised MNB model with Laplace smoothing prior. We use the MNB implementation in the scikit-learn library³.
- **SVM:** a strong baseline for various text classification baselines. We train a linear SVM using stochastic gradient descent with the default parameters in scikit-learn.
- **K-nearest neighbors (KNN):** we use a KNN model with a small $k=3$. Due to the large training data size, KNN's prediction is extremely slow, making it infeasible for production use.
- **ULMFiT:** a pioneer work successfully applying transfer learning to NLP (Howard and Ruder, 2018). The model was a previous state-of-the-art across multiple topic and sentiment classification tasks. We use the same parameters and training procedure following Howard and Ruder (2018).

³ <https://scikit-learn.org/>

Weakly-supervised learning baselines:

- **Generalized Expectation (GE):** uses labeled keywords to provide constraints that are optimized during training (Druck et al., 2008). We use the GE implementation in the MALLET library⁴.
- **MNB/Priors:** a weakly-supervised learning baseline that increases labeled keywords' prior and performs semi-supervised learning using EM algorithm (Settles, 2011). We use the implementation provided by the author⁵.
- **Doc2vec:** learns the document representation using a model that aggregates word embeddings belonging to the document (Le and Mikolov, 2014). We concatenate the keywords for a category to form a *pseudo document* and infer its embedding. During classification, we assign a document to a category with the nearest embedding. We use the doc2vec implementation in gensim⁶ with an embedding dimension of 100 and train the model for ten epochs.
- **WeSTClass:** a recent weakly-supervised neural text classification algorithm (Meng et al., 2018). We use the author's implementation⁷ and limit the supervision signal to keywords to compare with other weakly-supervised methods. We generate 500 pseudo documents for each category during pre-training and use the same unlabeled training corpus during self-training.

2.3.2.3 Performance metrics

We report the standard accuracy and Macro- F_1 scores for *news-crawl* and *news-crawl-v2* datasets. Macro- F_1 is more informative than Micro- F_1 because the categories are highly imbalanced.

We cannot apply standard multi-class classification metrics to *RTB* dataset because it is multi-label. Therefore, we calculate accuracy⁺ and $\text{ma}F_1$ following Nam et al. (2017). Accuracy⁺ is the proportion of correctly predicted labels. Since all models predict only one label, we count it as

⁴ <http://mallet.cs.umass.edu/>

⁵ <https://github.com/burrsettles/dualist>

⁶ <https://radimrehurek.com/gensim/>

⁷ <https://github.com/yumeng5/WeSTClass>

correct if the predicted label appears in the ground truth labels. maF_1 is a multi-label classification metric calculated as:

$$maF_1 = \frac{1}{L} \sum_{j=1}^L \frac{2tp_j}{2tp_j + fp_j + fn_j} \quad (5)$$

Where L denotes the number of labels and tp, fp, fn denotes the number of true positive, false positive, and true negative predictions. To achieve a perfect maF_1 of 1, the model needs to predict *all* the correct labels. Therefore, the models in comparison will obtain a score strictly lower than 1. However, the comparison is fair because they all predict exactly one label for each document.

2.3.3 Results and discussion

2.3.3.1 Mined keywords from labeled corpus

Table 3 presents the mined keywords for each category, which all weakly-supervised models use.

2.3.3.2 Text classification performance

Table 4 overviews various models' performance on the three evaluation datasets. We train all the supervised learning baselines with the complete labeled *news-crawl* training set. On the other hand, weakly-supervised learning models use the same dataset without labels.

The supervised learning baselines achieved better performance than the strongest weakly-supervised learning model on the *news-crawl* test set. However, their performance degraded drastically on out-of-domain evaluation datasets, including *news-crawl-v2*, which also consists of news articles. It underlines supervised models' inability to generalize to unseen data from a different distribution. Interestingly, ULMFiT achieved superior accuracy of 0.922 on the *news-crawl* test set, outperforming all other models by a large margin. Its performance is better than other supervised learning baselines on the other two datasets but lags behind weakly-supervised methods.

It is common for SVM to achieve better performance than MNB, as we observed on the *news-crawl* test set. Nevertheless, MNB had better generalization performance on the other two datasets. We conjecture it is because MNB models the data distribution from a generative

perspective instead of focusing on specific clues as in a discriminative model like SVM. KNN's performance on *news-crawl* dataset is inferior to other supervised baselines yet still reasonable. However, it failed on out-of-domain datasets, suggesting that it is most vulnerable to data drifts by calculating similarity scores with seen training examples.

Category	Mined keywords
Business	<i>aircraft railways ridership airframe airbus commuters aviation harvesting railroads roofing marketers boeings</i>
Society	<i>skoutmatchcom okcupid friendships transgender samesex lesbian marriages flirt lgbt dating lesbians heterosexual</i>
Technology and computing	<i>android scan apps firmware samsung os leftright device smartphones keyboard snapdragon 64bit usb smartphone</i>
Health and fitness	<i>symptoms inflammation medications disease vitamin disorders diabetes diet chronic diagnosis nutrition infections</i>
Law, government, and politics	<i>immigration passport uscis embassy attorney lawyers consular citizenship consulate lawyer legal citizens immigrants</i>
Science	<i>horoscopoe astrology atoms earths jupiter planets nasa molecules electrons telescope particles forecast orbit</i>
Sports	<i>olympics medal league semifinal finalsmidfielder freestyle championship semifinals football stadium athletes</i>
Art and entertainment	<i>bollywood actress actor films film song album singer actors songs lyrics comedy costar drama movie hollywood</i>
Education	<i>colleges universities students examacademic undergraduate admissions faculty examination cbse campus education</i>
Personal finance	<i>stocks investors securities nasdaq equity dividend investor bse earnings trading nse volatility bluechips intraday</i>
Automotive	<i>torque tires honda brakes wheels v8 exhaust transmission chevrolet steering engine cylinder dealership mileage sedan</i>
Food and drinks	<i>recipe sauce bake preheat recipes flour butter delicious flavor ingredients vanilla baking cheese stir garlic</i>
Family and parenting	<i>babys babycenter pregnancy babies trimester baby uterus pregnant breastfeeding placenta midwife newborn</i>
Style and fashion	<i>calories tattoo weightloss fat waistline dieting menswear acne sneaker carbs cardio dresses slimming moisturising</i>
Travel	<i>kayak booking rentals airline hotels attractions beaches resorts reservation reservations couchsurfing hotel</i>
Hobby and interest	<i>minecraft armor gameplay quests puzzle ingame multiplayer rpg enemies crossword weapons pokemon monsters</i>
Pets	<i>puppies vet puppy breeds dogs veterinarian breed dog pups breeders kennel pet terrier cats canine</i>
Religion and spirituality	<i>christians christ jesus bible religious worship islam christianity quran muslims church prayer scriptures muslim</i>
Home and garden	<i>diy wood soil gardeners cabinets backsplash mulch planting compost plants fertiliser decor watering screws potting</i>
Real estate	<i>furnished rent condo bedrooms rental sqft apartments apartment bedroom spacious trulia renovated vrbo rentals</i>
Careers	<i>vacancies recruitment candidates interviewer resume qualification employers employer freshers vacancy interviewers</i>
Shopping	<i>coupons coupon pricepony discount scoopon cashback freebies storewide</i>

Table 3: Mined keywords using *pmi-freq*.

It is noteworthy that while weakly-supervised models do not perform as well on the in-domain test set, they tend to perform robustly on out-of-domain datasets. It validated our motivation to achieve better transferability by abstracting the semantics via keywords.

Among the weakly-supervised methods, STM achieved the best performance on all three datasets, followed by GE, whose performance is consistently 1-3% lower. We note that both models explicitly utilize the word co-occurrence information, suggesting that it might be crucial for weakly-supervised models.

Model	News-crawl test set		News-crawl-v2		RTB data set	
	Accuracy	Macro- F_1	Accuracy	Macro- F_1	Accuracy ⁺	ma F_1
Random	0.045		0.045		0.067	
Most frequent	0.217		0.055		0.146	
MNB	0.817	0.766	0.524	0.466	0.660	0.504
SVM	0.850	0.811	0.489	0.470	0.471	0.381
KNN	0.751	0.679	0.189	0.159	0.166	0.103
ULMFiT	0.922	0.892	0.541	0.496	0.564	0.431
GE	0.510	0.483	0.596	0.587	0.777	0.617
MNB/Priors	0.533	0.411	0.439	0.366	0.631	0.493
Doc2vec	0.391	0.383	0.480	0.461	0.557	0.424
WeSTCLass	0.187	0.163	0.190	0.158	0.177	0.121
STM	0.544	0.527	0.623	0.607	0.794	0.625

Table 4: Each Model’s performance on three evaluation datasets. The best results for each metric are in boldface.

Surprisingly, WeSTCLass’s performance was very poor. Meng et al. (2018) experimented on binary sentiment classification and topic classification with few categories. However, our classification task contains 22 categories. A central assumption of WeSTCLass is that the keywords and documents related to each category lie in disjoint clusters in the embedding space. We conjecture that the larger number of categories caused the embeddings to overlap, thus decreasing the effectiveness of both the pseudo document generation and self-training steps.

The strong generalization performance of STM prompts us to ask the following questions:

1. Why STM’s performance on *news-crawl* test set is not competitive against supervised learning baselines?
2. What caused STM to yield a more robust performance on out-of-domain datasets?

We first plot STM’s confusion matrix on *news-crawl* test set in Figure 5. The “misclassifications” do not seem random, but among closely related categories such as “Business” and “Personal finance.” Other cases such as misclassifying “Pets” to “Family and parenting” are worth investigating.

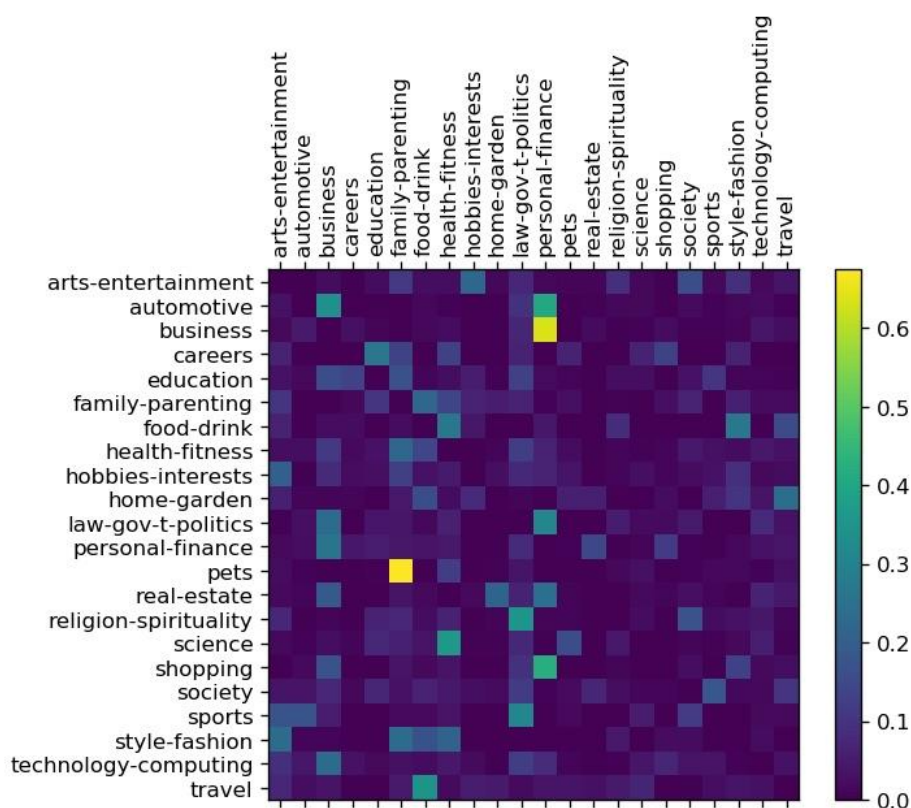


Figure 5: STM’s prediction confusion matrix on *news-crawl* test set. We removed the diagonal entries (correct predictions) to surface misclassifications.

Table 5 shows the most frequent predictions STM predicted for documents belonging to the “Pets” category. Our intuition is that some of these categories are related to pets in that pets are an essential part of a family. In addition, some articles might also mention veterinary medicine, thus causing the model to predict “Health and fitness.”

Category	No. of predictions
Pets	194
Family and parenting	89
Health and fitness	16
Law, government, and politics	5
Travel	4

Table 5: Five most frequent categories STM predicted for documents in the pets category.

Table 6 shows examples where STM predicts “Family and parenting” for documents belonging to the “Pets” category. These documents seem to relate to both categories. Because the *news-crawl* dataset assigns only one label per document, there might be plausible labels related to the document that are not in the ground truth.

#	Text
1	poll Do you think it's important for children to have pets? No, pets only make messes Yes, it teaches them responsibility. Share your vote on facebook so your friends can take this poll
2	Babies versus pets in viral advertising posted which do you prefer? Pets or babies? They're everywhere in social media pulling views sparking massive followings rising to the top of every hit list it's a massive love fest huh? what's going on? have we gone cute crazy? why do these characters work so well? . . .
3	'The dog is (by which you mean, 'I want a divorce!') . . . The dog is bored is my husband projecting? transferring? planning on taking the dog for a romantic tropical vacation? Am I right? Am i crazy? You decide. Relationships are full of mystery and are open to interpretation, wild speculation and deep neurosis . . .

Table 6: Sample documents with label ‘Pets’ that STM classifies as “Family and parenting.”





Figure 6: LIME explanations on a sample where both models predicted the correct label.

To answer the second question, we employ LIME (Ribeiro et al., 2016), a model-agnostic interpretation toolkit, to visualize what features STM and ULMFiT utilize while performing predictions. LIME perturbs the input text and trains a local linear classifier, from which it extracts the feature (word) importance towards the classification result.

Figure 6 and Figure 7 depict two sample documents and the LIME interpretation for both models. Both models predict correctly for the example in Figure 6. However, STM focused on discriminative keywords while ULMFiT attended to irrelevant words like “reddit” and “your.” On the other hand, STM predicted the correct category with plausible clues in Figure 7, but ULMFiT predicted the wrong label. Overall, ULMFiT uses “fuzzier” features, a common characteristic of black-box deep learning models. While these features might help the model fit the training

distribution perfectly, they do not generalize well to new domains and thus have a poor generalization performance.

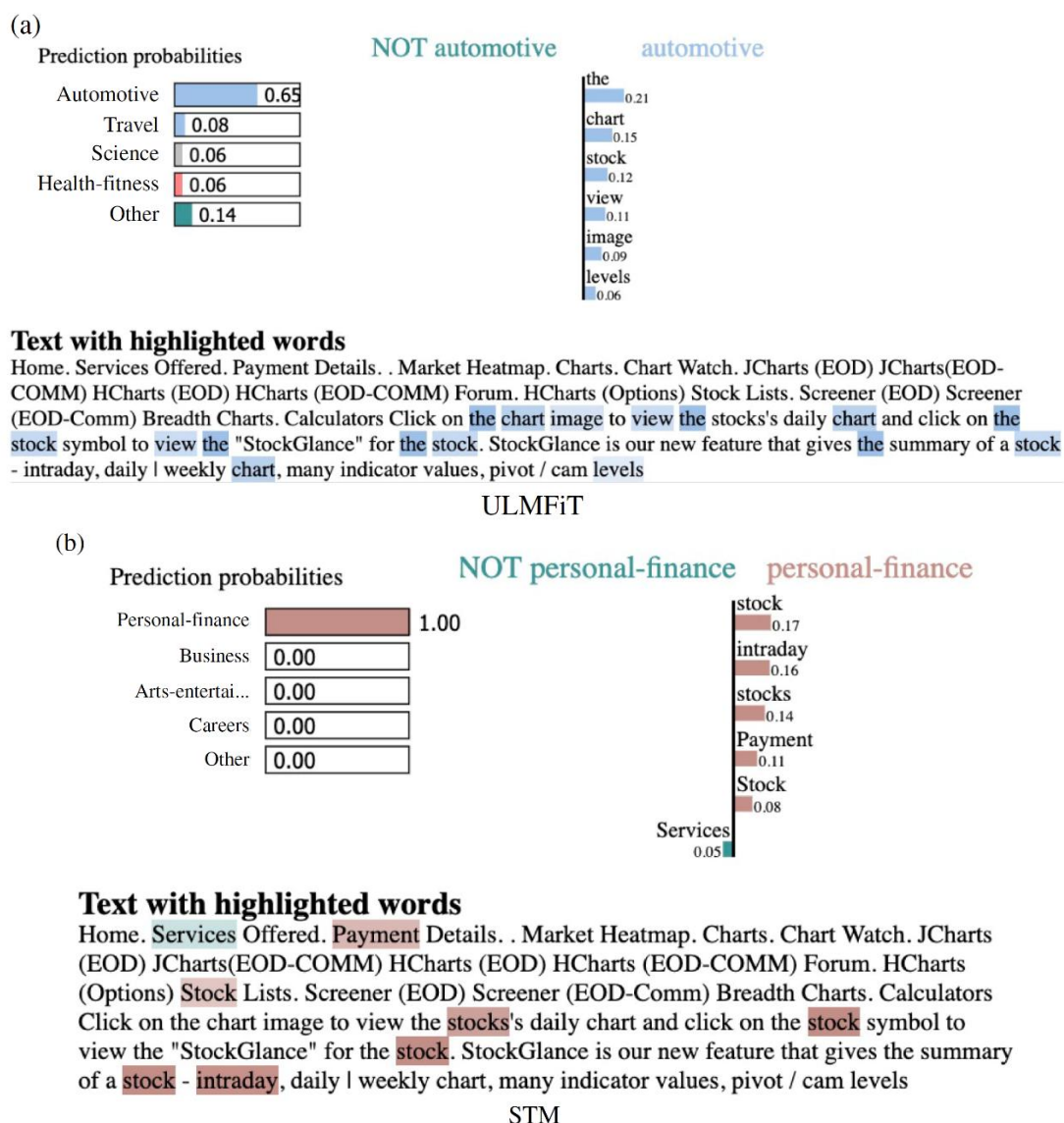


Figure 7: LIME explanations on a sample document where STM predicted correctly, but ULMFiT predicted wrongly.

2.3.3.3 Impact of keyword selection strategy

A crucial contribution of this work is a robust keyword mining algorithm to automatically mine keywords for weakly-supervised models. We performed an intrinsic evaluation of the keyword quality in Section 2.2.1. We also wish to measure the impact of different sets of keywords on the

final accuracy of weakly-supervised models. Therefore, we trained various STM models with the keywords mined using the following algorithms⁸ and present the performance in Table 7:

- ***label***: uses only the words appearing in the category name.
- ***freq***: selects the most frequent words appearing in each category, excluding stopwords.
- ***mi***: selects keywords using mutual information.
- ***pmi-freq***: our proposed method.

Model	News-crawl test set		News-crawl-v2		RTB data set	
	Accuracy	Macro- F_1	Accuracy	Macro- F_1	Accuracy ⁺	ma F_1
STM + $S_{pmi-freq}$	0.544	0.527	0.623	0.607	0.794	0.625
STM + S_{label}	0.270	0.243	0.332	0.259	0.405	0.340
STM + S_{freq}	0.284	0.257	0.425	0.359	0.500	0.358
STM + S_{mi}	0.301	0.265	0.434	0.344	0.565	0.385

Table 7: STM’s performance using different sets of keywords.

Among the baselines, using only the category name resulted in the worst performance, showing that the category name might be ambiguous and lacks expressive power compared to discriminative keywords. Using *freq* or *mi* to mine keywords outperformed using the category name alone, but they far lagged behind our proposed *pmi-freq* method. Based on the result of this section and the previous one, we highlight that keyword selection is at least as important as the choice of models. However, it has not received enough attention from the research community.

2.3.3.4 Domain adaptation performance

We demonstrated in Section 2.3.3.2 that STM achieved superior generalization performance even without using any in-domain data. Additionally, a key advantage of weakly-supervised learning methods is they can utilize *unlabeled* in-domain datasets. Therefore, we trained another STM model using the same seed keywords and 280k unlabeled web pages from the RTB traffic. We denote the model as “STM cross-domain” and compare its performance with the original model in Table 8.

⁸ The keyword selection methods are detailed in Section 2.2.1.

Model	News-crawl test set		Browsing data set	
	Accuracy	Macro- F_1	Accuracy ⁺	ma F_1
STM	0.544	0.527	0.794	0.625
STM cross-domain	–	–	0.814 (+2.5%)	0.647 (+3.5%)

Table 8: Domain adaptation performance using unlabeled in-domain data.

As expected, the new model performed better on the *RTB* test dataset, which is from the same distribution as the new unlabeled training data. In addition, it also performed better on *news-crawl-v2* dataset. Thus, we hypothesize that training on the heterogeneous RTB data leads to a model more robust to domain variations.

2.4 Conclusions

In this chapter, we introduced a novel framework to mitigate the problem of out-of-domain noisy training data in contextual targeting. We first mine keywords from the existing training data using a robust statistical method. We then train a weakly-supervised model using the mined keywords and unlabeled corpora. We demonstrated that utilizing an unlabeled in-domain dataset yielded a further 3% improvement in performance. Finally, we benchmarked our proposed method with various supervised and weakly-supervised learning methods and achieved superior generalization performance on two out-of-domain evaluation datasets.

3. Evaluating Weakly-Supervised Classifiers with Bayesian Error Estimation

While weakly-supervised classifiers can sometimes achieve comparable accuracy with fully-supervised classifiers, they rely on high-quality seed words (Li et al., 2018; Jin et al., 2020). Most such methods rely on explicit keyword bootstrapping. Therefore, the final classifiers’ accuracy can differ drastically depending on the input seed words. Furthermore, many seed words used in previous work are not intuitive. For example, Meng et al. (2019) used {united, champions, cup} for the category “soccer” instead of obvious seed words such as “football” or “soccer.” We conjecture that the authors might have experimented with the obvious seed words and replaced them with more discriminative words based on *the classifier’s performance*.

When conducting research work, we usually have a *labeled* dataset a priori, with which we can assess the performance of text classifiers using standard metrics such as accuracy and F_1 score. However, we do not have access to any labeled data when applying weakly-supervised classification methods in the real-world setting, neither for training nor for evaluation. It caused two significant challenges. Firstly, we cannot measure the classifier’s performance. Therefore we either have to conduct extensive manual testing or deploy the model blindly to production without knowing whether it “works.” Secondly, the absence of an evaluation or development dataset makes it impossible to perform hyperparameter tuning, the most crucial hyperparameter being the selection of seed words.

This chapter proposes *OptimSeed*, a novel framework to automatically mine and rank seed words directly based on their estimated accuracy. First, we mine seed words co-occurring with the *category name* using the method we proposed in Section 2.2.1. Then, we train *interim* classifiers with individual seed word pairs. Finally, we apply an unsupervised error estimation method to estimate the accuracy of the interim classifiers. We add the seed words that yield the highest estimated accuracy to the final seed word set. Figure 8 depicts our proposed framework. We evaluate our proposed framework on six binary classification tasks drawn from four datasets. *OptimSeed* outperforms a baseline using only the category name as the seed word and achieved comparable performance using expert-curated seed words reported in previous work. The main contributions of this work are as follows:

1. Applying unsupervised error estimation to weakly-supervised text classification to improve the classification accuracy and robustness.
2. Conducting a comprehensive study on the impact of different seed words across various classification tasks and models.

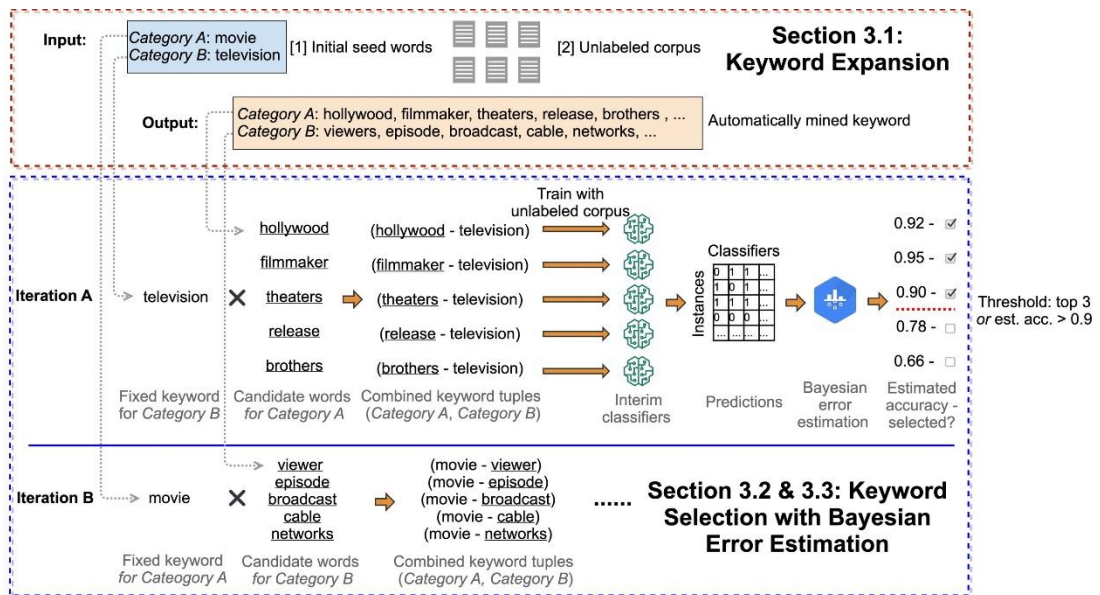


Figure 8: *OptimSeed*, a framework to select seed words for weakly-supervised text classification using unsupervised error estimation.

3.1 Related Work

3.1.1 Weakly-supervised text classification

We present additional previous work not discussed in Section 2.1.1 in this section.

Mekala and Shang (2020) proposed to use contextualized word representation to disambiguate different word senses of the seed words. They used a BERT (Devlin et al., 2019) model to derive the contextualized word representation of each *word occurrence*. Then they clustered word occurrences into different word senses using the k-means algorithm. Each word sense is treated separately to avoid ambiguous word senses. In their motivating example, the word “penalty” can occur in both “sport” and “law” contexts. Their method refines the seed word list and trains weakly-supervised classifiers iteratively. They simultaneously improve each other using self-training.

On the other hand, Meng et al. (2020) and Wang et al. (2021) tackled the same problem as our work: training text classifiers using *only* the category name. These works utilized contextualized word embeddings to bootstrap more relevant keywords like Mekala and Shang (2020).

3.1.2 Unsupervised error estimation

Traditionally, we need a dataset with gold-standard labels to measure the accuracy of classifiers. Unsupervised error estimation tries to estimate a list of classifiers' error rates with *unlabeled* data. To our best knowledge, all previous work in weakly-supervised classification relied on labeled datasets to conduct evaluations. While they allow easy comparison, as we mentioned earlier, we cannot assume the availability of labeled evaluation datasets in the real-world weakly-supervised classification setting.

Most previous work in unsupervised error estimation derives the analytical error rate with some simplifying assumptions. For example, Donmez et al. (2010) and Jaffe et al. (2015) assumed that the true marginal category probability $p(y)$ is known. Likewise, Platanios et al. (2014) assumed each classifier makes conditionally independent errors. These approaches provided an essential theoretical foundation. Unfortunately, such assumptions do not hold in the real-world scenario, casting doubts on whether the analytical error rates are reliable.

Platanios et al. (2016) introduced Bayesian error estimation. The model performs Gibbs sampling and jointly infers the true category and the error rates. The approach demonstrated superior performance compared to several baselines. The estimated error rates usually lie within a few percent away from the actual value. Additionally, they only make a mild assumption that more than half of the classifiers have an error rate lower than 50%.

3.1.3 Keyword mining

Keyword mining algorithms take a small list of seed words and bootstrap high-quality keyword lexicons. It was widely employed in opinion mining (Hu and Liu, 2004; Hai et al., 2012) and technical term mining (Elhadad and Sutaria, 2007). Much previous work in weakly-supervised classification also involves keyword mining (Meng et al., 2020; Wang et al., 2021) as an internal step. The additional step in weakly-supervised text classification is to use the mined keywords to iteratively refine the classifiers. In a sense, keyword mining and weakly-supervised text

classification both take a small list of seed words and an unlabeled dataset and “expand” the knowledge about the target category. More discriminative keywords will aid the classification accuracy, while an accurate classifier produces much less noise in pseudo-labeling and makes keyword mining much more straightforward.

3.2 Proposed Method

3.2.1 Mining candidate keywords from a single seed word

The upper box in Figure 8 shows the keyword mining step of *OptimSeed*. To ensure reproducibility, we purposely *avoided* carefully choosing the initial seed word. Instead, we use either the category name or trivial keywords (e.g., “good” and “bad” for sentiment classification) as the only input seed words. We use *pmi-freq*, the same method we introduced in Section 2.2.1, to rank all words in the vocabulary based on their association with the input seed word s .

$$pmi - freq(w; s) \equiv \begin{cases} \log df(w) \cdot pmi(w; s), & \text{if } df(w) \geq 5 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Furthermore, we filter the candidate words based on their part-of-speech tag. We keep only adjectives for sentiment classification and nouns for topic classification. We then select the top k candidates for each category based on their *pmi-freq* score.

3.2.2 Training interim classifiers

We utilize mined candidate keywords and an *unlabeled* dataset to train *interim classifiers*. We use a single seed word for each category to train interim classifiers to isolate the impact of each seed word. As illustrated in Figure 8 Iteration A, we combine the category name for category B (Television) with each candidate seed word of category A (Movie) to form seed word tuples, which are the input to the corresponding interim classifiers.

We use Generalized Expectation (GE) (Druck et al., 2008) as the underlying weakly-supervised classification algorithm for interim and final classifiers. As demonstrated in Section 2.3.3.2, GE performed consistently well, only lagging behind STM by a small margin. However, STM involves heavy indexing and inference computation. For the same classification task, STM usually takes several hours to train while GE takes only a few seconds. Because we need to train a

relatively large number of interim classifiers, using STM would yield unacceptable computation and latency.

Underlying, GE translates each labeled seed word to a constraint function. For example, the keyword tuple (filmmaker, television) translates to two constraints: filmmaker \rightarrow A: 0.9, B: 0.1 and television \rightarrow A: 0.1, B: 0.9, meaning the word “filmmaker” would appear 90% in a document belonging to category A while 10% in a document belonging to category B, vice versa for the keyword “television.”

The constraints on each labeled keyword w_k then add up to form the objective function in the equation below:

$$\delta = - \sum_{k \in K} L^2(\hat{p}(y|w_k) || \tilde{p}(y|w_k)) \quad (7)$$

Where $\hat{p}(y|w_k)$ is the reference category distribution defined by the constraint function, and $\tilde{p}(y|w_k)$ is the empirical category distribution predicted by the classifier. The model is trained by minimizing the L2 distance $\sum_i (\hat{p}(y_i|w_k) - \tilde{p}(y_i|w_k))^2$ between the two distributions, which sums over the difference in probability over each category i .

3.2.3 Evaluating seed words with Bayesian error estimation

With a list of interim classifiers and their predictions on an unlabeled “development” dataset, we can apply unsupervised error estimation to estimate the accuracy of each interim classifier. The seed words used in the best-performing interim classifiers are then selected for the final classifier. As demonstrated in Figure 8 Iteration A, the three candidate words “hollywood”, “filmmaker”, “theaters” are chosen based on the interim classifiers’ accuracy. We repeat the procedure to select seed words for category B in the subsequent iteration.

Based on the literature survey, we chose Bayesian error estimation (BEE) (Platanios et al., 2016) to estimate the error rates. As its name suggests, BEE employs a Bayesian perspective where each instance’s label l_i and each classifier’s error rate e_j are latent variables. The model first infers the true label l_i for each instance based on the predictions of interim classifiers, then uses the inferred label to calculate the error rate of each interim classifier e_j .

BEE performs inference using Gibbs sampling, depicted in Figure 9. We provide the details of the conditional probabilities as follows. Firstly, we draw $p \sim \text{Beta}(\alpha_p, \beta_p)$, the prior probability for the true label being equal to 1 over the whole dataset S :

$$P(p|\cdot) = \text{Beta}(\alpha_p + \sigma_l, \beta_p + S - \sigma_l) \quad (8)$$

Then for each data point, we draw a label $l_i \sim \text{Bernoulli}(p)$:

$$P(l_i|\cdot) = p^{l_i}(1-p)^{1-l_i}\pi_i \quad (9)$$

We further assume an underlying distribution of error rates for the interim classifiers, f_j . We draw an error rate $e_j \sim \text{Beta}(\alpha_e, \beta_e)$:

$$P(e_j|\cdot) = \text{Beta}(\alpha_e + \sigma_j, \beta_e + S - \sigma_j) \quad (10)$$

Finally, for each example i and classifier j , we make a prediction by:

$$\hat{f}_{ij} = \begin{cases} l_i, & \text{with probability } 1 - e_j, \\ 1 - l_i, & \text{otherwise,} \end{cases} \quad (11)$$

Where:

$$\sigma_l = \sum_i^S l_i, \quad \sigma_j = \sum_i^S \mathbb{1}_{\{\hat{f}_{ij} \neq l_i\}}, \quad \pi_i = \prod_{j=1}^N e_j^{\mathbb{1}_{\{\hat{f}_{ij} \neq l_i\}}} (1 - e_j)^{\mathbb{1}_{\{\hat{f}_{ij} = l_i\}}} \quad (12)$$

We highlight the dependency between the two variables: $e_j \rightarrow \pi_i \rightarrow l_i$ and $l_i \rightarrow \sigma_l \rightarrow \sigma_j \rightarrow e_j$. Therefore, e_j and l_i are *interdependent*. We obtain joint inference of the two variables by performing Gibbs sampling for multiple iterations till convergence.

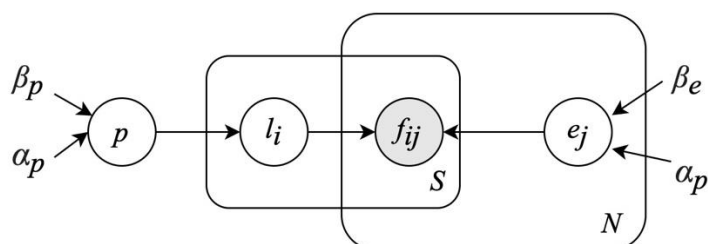


Figure 9: Graphical model for error estimation.

3.3 Experiments

3.3.1 Experimental setup

We benchmark our framework using six binary classification tasks drawn from four popular text classification datasets. We select the evaluation tasks to ensure coverage of different domains and granularities. We provide the details as follows:

- **AG’s News Dataset:** containing 120k documents evenly distributed into four coarse-grained categories. The documents are short texts corresponding to the first paragraph of the news articles. We randomly picked two binary classification tasks: “Politics” vs. “Technology” and “Business” vs. “Sports”.
- **The New York Times (NYT) Dataset:** containing 13k documents covering 25 fine-grained categories. The documents are long texts of full news articles. We selected two binary classification tasks involving similar categories: “International Business” (InterBiz) vs. “Economy” and “Movies” vs. “Television”.
- **Yelp Restaurant Review Dataset:** containing 38k evenly distributed reviews belonging to 2 categories: “Positive” vs. “Negative”.
- **IMDB Movie Review Dataset:** containing 50k evenly distributed reviews belonging to 2 categories: “Positive” vs. “Negative”.

We compare against the following weakly-supervised classification methods:

- **Dataless (Chang et al., 2008):** using Explicit Semantic Analysis (ESA) (Gabrilovich et al., 2007) to map categories and documents into a shared semantic space. Then perform classification by nearest neighbor search.
- **MNB/Priors (Settles, 2011):** increasing labeled keywords’ prior and performing semi-supervised learning with half-step of EM algorithm to induce the classifier.

- **WeSTClass (Meng et al., 2018):** first generating pseudo documents using a generative model, then refining the model with self-training. We employ the CNN architecture following Meng et al. (2018).
- **ConWea (Mekala and Shang, 2020):** differentiating different senses of seed words using contextualized embeddings and training classifiers and expanding seed words iteratively.

Besides, we also compare with a fully-supervised logistic regression (LR) model trained using all labeled documents in the training dataset.

Category	Mined keywords	Previous Work
Politics	<i>political;</i>	<i>democracy religion liberal;</i>
Tech	<i>technology</i>	<i>scientists biological computing</i>
Business	<i>business;</i>	<i>economy industry investment;</i>
Sports	<i>sports</i>	<i>hockey tennis basketball</i>
InterBiz	<i>international;</i>	<i>china union euro;</i>
Economy	<i>economy</i>	<i>fed economists economist</i>
Movies	<i>movie;</i>	<i>hollywood directed oscar;</i>
Television	<i>television</i>	<i>episode viewer episodes</i>
Yelp & IMDB	<i>good;</i> <i>bad</i>	<i>terrific great awesome;</i> <i>horrible subpar disappointing</i>

Table 9: Initial seed words for each classification task.

We mine sixteen candidate seed words for each category across all experiments. We select a seed word for the final classifier if its estimated accuracy is higher than 90% or among the top three seed words based on the estimated accuracy. We use a reference distribution of 90%/10% (signifying that a labeled seed word would occur 90% in a document of the specified category and 10% in another category). Table 9 compares the initial seed words used in our work and previous work⁹.

3.3.2 Classification performance

Table 10 presents the accuracy for topic classification tasks. OptimSeed achieved competitive and consistent performance across all models and tasks. It outperformed using the category name

⁹ For the seed words used in previous work, NYT corpus were from Meng et al. (2019) and the rest are from Meng et al. (2018). IMDB corpus has not been used in previous work in weakly-supervised text classification. Therefore, we use the same seed words as the Yelp dataset.

as seed words 80% of the time (16/20) while outperforming the expert-curated seed words 65% of the time.

ConWea often fails when the input seed word quality is low (the “cate” column) despite its claim to disambiguate word senses. On the Biz-Sport task, its accuracy was only 39.1%. We manually examined the keyword sets expanded by ConWea and found them much noisier than our proposed framework.

Method	Poli-Tech			Biz-Sport			IB-Econ			Movie-TV		
	cate	ours	gold	cate	ours	gold	cate	ours	gold	cate	ours	gold
Dataless	50.1	51.4	50.2	50.0	50.2	50.4	59.1	75.0	67.1	67.8	70.0	67.8
MNB/Priors	87.3	88.9	88.9	95.6	93.9	92.9	58.5	54.3	93.9	67.8	67.8	68.9
W _{EST} CLASS	87.4	89.5	88.8	92.7	94.8	94.3	77.7	83.0	75.1	50.4	76.6	62.1
ConWea	71.5	73.7	71.4	39.1	67.0	82.0	75.1	71.2	84.3	66.9	77.0	76.4
GE	86.9	87.8	88.5	93.0	93.0	79.4	70.7	81.7	91.5	94.4	98.9	97.8
LR		96.3			98.6			90.2			85.5	

Table 10: Accuracy on topic classification tasks. *cate*, *ours*, *gold* represent using the category name, OptimSeed seed words, and human-curated seed words reported in previous work. We highlight the best-performing keyword set for each model.

Table 11 summarizes the performance for sentiment classification tasks. Again, we can observe similar trends as in topic classification. However, the gap between weakly-supervised classifiers and LR is much more substantial, suggesting more nuance in expressing various sentiments.

Method	Yelp			IMDB		
	cate	ours	gold	cate	ours	gold
Dataless	51.0	55.5	52.2	50.1	60.4	52.2
MNB/Priors	50.9	71.5	51.7	51.1	54.0	50.3
W _{EST} CLASS	78.3	58.8	81.5	67.7	60.6	60.5
ConWea	51.0	51.3	50.7	56.5	55.7	59.1
GE	68.0	75.2	79.3	69.6	72.2	74.0
LR		92.2			88.3	

Table 11: Accuracy on sentiment classification tasks. We highlight the best-performing seed words in bold for each model-task combination.

Table 12 averages the performance over six classification tasks. OptimSeed seed words performed better than using the category name alone by a large margin for all models. It validates that keyword expansion is essential in improving the performance of weakly-supervised

classification methods. Furthermore, when applying the GE learning algorithm, OptimSeed’s average accuracy is only 0.3% lower than using human-curated seed words. It can potentially eliminate human experts from the loop.

Method	cate	ours	gold
Dataless	54.7	60.4*	56.7
MNB/Priors	68.5	71.7	74.4
W _{ESTCLASS}	75.7	77.2	77.0
ConWea	60.0	66.0	70.7
GE	80.4	84.8*	85.1
LR		91.8	

Table 12: Average accuracy scores over six classification tasks. We highlight the best performing seed words for each model in bold. * denotes statistical significance using double-sided paired T-test with a p-value of 0.05 w.r.t. the same model using the “cate” seed words.

3.3.3 Case study

We present a case study of the “International Business” vs. “Economy” classification task to demonstrate the working our OptimSeed. Table 13 presents seed words sets at different stages for the “Economy” category.

Keyword expansion helped to improve the accuracy significantly over the initial seed word. However, it runs the risk of introducing noises. The Bayesian error estimation step helps the model to focus on reliable seed words like “economist” and “economists” while eliminating ambiguous seed words such as “growth” and “purchases”. The step helped to improve the accuracy by a further 2.4%.

Stage	Seed Words for “Economy”	Accuracy
Initial	<i>economy</i>	70.7
Keyword Expansion	<i>purchases pace index borrowing unemployment economists</i> <i>economy stimulus rates recovery economist rate fed reserve</i> <i>inflation growth</i>	79.3
Final	<i>economist economists rate recovery index</i>	81.7

Table 13: Seed words for “Economy” at various steps within the OptimSeed framework.

3.4 Conclusions

This chapter introduced *OptimSeed*, a novel framework to perform unsupervised evaluation and hyperparameter tuning (in terms of choosing seed words) for weakly-supervised text

classification. It yielded robust performance across various models and datasets and matched expert-curated seed words. We believe our framework will facilitate monitoring and evaluation of weakly-supervised models in production without needing any manually labeled data.



4. Generating Slogans with Seq2seq Transformers

4.1 Related Work

4.1.1 Slogan generation

Slogans play an essential role in advertising and are characterized by being concise and catchy. Traditionally, it is human copywriters' task to compose slogans. Copywriting demands creativity and in-depth knowledge about the advertised product or service. Previous work focused on altering existing slogans and automatically generating slogans by inserting novel keywords or concepts into slogan skeletons to speed up the process.

BrainSup (Özbal et al., 2013) is one of the first frameworks for creative sentence composition, allowing users to enforce specific keywords or emotion constraints. It first mines morpho-syntactic patterns from a dependency-parsed dataset. The patterns contain several empty slots and are used as skeletons to compose new slogans. During the generation phase, the framework narrows down to a set of skeletons compatible with the user-input constraints. It then tries to fill in the slots and rank the candidates based on how well they satisfy the user input.

Inspired by Özbal et al. (2013), Tomašić et al. (2014) also utilized slogan skeletons specified by POS tags and dependency types. The input to their algorithm is a company or product description. Instead of letting the users manually specify keywords, the system extracts keywords and entities automatically. Additionally, they used genetic algorithm to sufficiently explore the search space. The genetic algorithm's initial population is generated using random skeletons. Tomašić et al. (2014) created a list of ten heuristic-based scoring functions to evaluate each population generation and applied crossovers and mutations to produce a new generation. Specifically, mutation implies replacing a random word with another word with the same POS tag. Crossover picks a random word pair from two slogans and flips them. E.g., input: ["*Drink* more milk", "Just *do* it"] → ["*Do* more milk", "Just *drink* it"].

Gatti et al. (2015) aimed to "refresh" well-known expressions by infusing novel concepts from news articles. They first detect keywords from news articles and expand them using knowledge bases. They then inject the keywords into BrainSup-style skeletons. They also introduced a word embedding validation. Namely, the keywords blended into a well-known expression must have a similar embedding as the words they replace. This validation helps to eliminate nonsense output like "Do more milk." The final step involves scoring all candidates based on the word embedding

similarity and dependency scores, trading off relatedness and grammaticality. Gatti et al. (2017) also applied a similar approach to modify song lyrics in another work.

Iwama and Kano (2018) introduced a slogan generation system utilizing a slogan database, case frames, and word embeddings. The system produced slogans with impressive quality and was deployed to production by one of the world’s largest advertising agencies. However, the system requires manual selection from a pool of ten times large samples. Therefore, we cannot conclude whether the superior quality is due to the system or human curation. Furthermore, Iwama and Kano (2018) did not disclose the details of their system, making it impossible to reproduce their result.

Most recently, Alnajjar and Toivonen (2021) focused on generating nominal metaphors for slogan generation. A nominal metaphor involves a target concept T (e.g., car) and a property P describing the target concept (e.g., elegant). The system aims to generate metaphors that bind the target concept and the property (e.g., “The Car Of Stage”). Underlying, the system searches for candidate metaphorical vehicles¹⁰ v given a target concept T and a property P . It then searches for fillable slots for each skeleton s and the $\langle T, v \rangle$ pair. Finally, the system synthesizes candidate slogans by filling in the slots and optimizing with genetic algorithm following Tomašić et al. (2014).

Apart from skeleton-based approaches, recent advances in neural language models enabled research in generating slogans (from scratch) without any template. Munigala et al. (2018) tackled the problem of generating persuasive sentences in the fashion domain. They first extract domain-specific keywords from input product specifications, then expanded them with relevant noun phrases and verb phrases. The system synthesizes sentences from the keywords using a large domain-specific neural language model (LM). Munigala et al. (2018) restricted the LM’s vocabulary to the extracted keywords, in-domain noun phrases, verb phrases, and frequent function words. The LM is trained by optimizing the overall perplexity using beam search. The sentences always begin with a verb to form imperative sentences because they are more assertive and persuasive. It is important to note that Munigala et al. (2018)’s approach does not rely on labeled data or parallel text. They demonstrated that their *unsupervised* approach outperformed a

¹⁰ A metaphor has two parts: the tenor (target concept) and the vehicle. The vehicle is the object whose attributes are borrowed.

supervised LSTM encoder-decoder model. However, the encoder-decoder model was trained on a much smaller parallel corpus than the corpus they trained the domain-specific language model.

Misawa et al. (2020) addressed slogan generation using a Gated Recurrent Unit (GRU) encoder-decoder model. They highlighted that a good slogan should be distinctive w.r.t. the target product. Therefore, they introduced a reconstruction loss to improve distinctiveness and employed a copying mechanism to deal with out-of-vocabulary words in the input sequence (e.g., product names). Their final model achieved a best ROUGE-L score of 19.38, outperforming strong encoder-decoder baselines.

Our work is most related to Misawa et al. (2020) because we also apply an encoder-decoder model. However, we differ from Misawa et al. (2020) in two main aspects. Firstly, we use a more modern Transformer architecture (Vaswani et al. (2017)), the current state-of-the-art for most language generation tasks. The powerful Transformer model enhanced with pretraining is more flexible and versatile than a recurrent neural networks model. We do not face the problem of generating generic slogans and out-of-vocabulary words (thanks to sub-word tokenization). Therefore, we can use the standard encoder-decoder Transformer model and train it using a cross-entropy loss. Secondly, we propose novel approaches to improve the coherence and diversity of generated slogans, validated by comprehensive automatic and human evaluation.

4.1.2 Sequence-to-sequence models

Sequence-to-sequence (seq2seq) models, also known as encoder-decoder models, are well-suited for conditional generation tasks where the model takes an input sequence and generates an output sequence. In contrast to vanilla language models, there is no one-to-one correspondence between the input and output tokens. Sutskever et al. (2014) presented a seminal work extending Long Short-Term Memory (LSTM) models to seq2seq tasks. Their method encodes the input sequence into a fixed-dimension hidden vector, then decodes the entire output sequence conditioned on the vector. Sutskever et al. (2014) applied the model to the English-French translation task and achieved near state-of-the-art (SOTA) performance. The result was astounding because a large research team developed the previous state-of-the-art (statistical) machine translation model over multiple decades, while a handful of researchers developed the new model within several months.

An obvious bottleneck of Sutskever et al. (2014) is the fixed dimensionality of the hidden vector. Conceptually, all the information in the input sequence has to be “compressed” into the vector. While it is possible for shorter sequences (< 20 tokens), the performance degrades drastically for longer sequences. Luong et al. (2015) and Bahdanau et al. (2015) proposed the attention mechanism to address this issue. Instead of compressing the input sequence into a single hidden vector, the model stores the contextualized representation at each time step. Then, when performing decoding, the model aggregates the contextualized representation dynamically based on the previously decoded partial sequence. Seq2seq models with attention outperformed previous SOTA on English-German and English-French translation tasks and was soon popularized in almost all NLP subtasks.

Despite LSTM’s impressive performance, it can only compute the hidden states one step at a time because the hidden state of time step t depends on the previous time step $t-1$. Therefore, LSTM cannot be fully parallelized on modern GPUs and often take several weeks to train. Vaswani et al. (2017) introduced a new architecture, the Transformer, solely relying on multi-head self-attention blocks. The Transformer models initially received skepticism because the sequential nature of recurrent neural networks appeals better to the intuition of human language. However, Transformers achieved strong empirical results by attaining a new SOTA with a shorter training time.

Transformer models are unidirectional in that every token only attends to the representation of previous tokens in the self-attention layer. It is partially due to the auto-regressive language model learning objective. BERT (Devlin et al., 2019) improved the Transformer model by allowing it to capture bidirectional context. BERT is trained using a novel masked language model (MLM) learning objective by randomly masking tokens in the input sequence and predicting it given the surrounding tokens. More importantly, Devlin et al. (2019) popularized transfer learning for NLP by separating training into two stages: pre-training using unsupervised learning objectives and task-specific fine-tuning.

Underlying, BERT is the encoder portion of the Transformer model. While it achieved SOTA results on language understanding benchmarks, it cannot perform auto-regressive generation naturally. To address this limitation, Lewis et al. (2020) presented BART, an encoder-decoder Transformer model combining a BERT-style bidirectional encoder and an auto-regressive

decoder. Furthermore, Lewis et al. (2020) introduced novel pre-training objectives well suited for seq2seq tasks, such as masking arbitrary text spans, deleting tokens, shuffling sentences, and rotating documents.

Transformer language models pre-trained on large corpora can often generate realistic-looking text. However, users do not have much control over the topic or style of the generation. Keskar et al. (2019) introduced CTRL, a conditional transformer language model that controls the style and content using control codes. Control codes are short tokens that encode the topic, domain, or sentiment. Formally, by conditioning on the control codes c during pre-training, CTRL can mix and match control codes and generate novel text at inference time with the conditional probability $p(x_i | x_{<p} c)$.

This work follows a BART model architecture due to its strength for seq2seq tasks. Inspired by CTRL, We also use control codes to generate syntactically-diverse slogans.

4.2 Datasets

Modern seq2seq models require a sizeable parallel dataset to train. While large advertising agencies might have run hundreds of thousands of ad campaigns and possess sufficient internal data (Kanungo et al., 2021), such data is not available to the research community due to data privacy concerns.

Instead, we crawl (description, slogan) pairs from a large publicly-available company dataset. The Kaggle 7+ Million Company Dataset¹¹ contains 7 million records of company information, including name, industry, size, URL, etc. Most companies include the “description” field in their HTML page’s `<meta>` tag. It constitutes the input to our model. Many companies also include their slogan in the HTML page title, such as “GoPro | World’s Most Versatile Cameras | Shop Now & Save.” We use various keywords, lexicographical, and semantic rules to filter and clean the HTML page title to obtain the slogans (In the previous case, the output is “World’s Most Versatile Cameras”).

Out of the 7M company records, we could crawl 1.4M companies with both the meta tag description and HTML title. After cleaning, we obtained 340k (description, slogan) pairs. We split 2% of the data for validation and test set each. We further curated a random sample of 1,467

¹¹ <https://www.kaggle.com/peopledatalabssf/free-7-million-company-dataset/>

raw slogans from the test set because the dataset was constructed using automatic data cleaning and may contain noise. Our final validation set and curated test set contain 5,412 and 1,000 (description, slogan) pairs each.

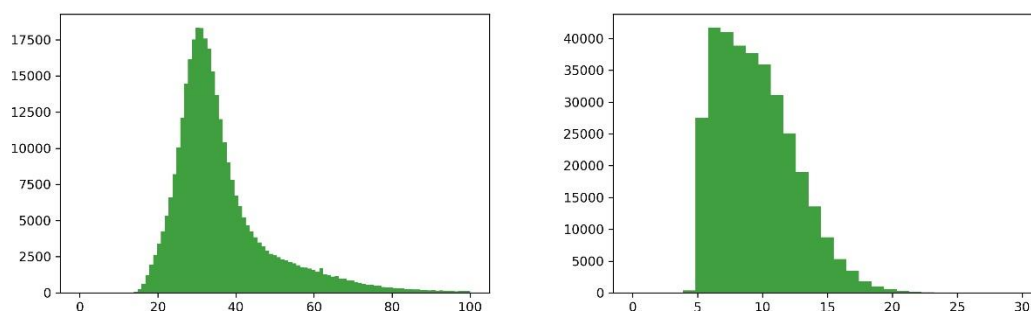


Figure 10: Distribution of the number of (subword) tokens. Left: description. Right: slogan.

We use BART’s tokenizer to tokenize both the descriptions and the slogans. Figure 10 overviews the distribution of the token length. We can observe that the slogans are usually very concise and have fewer than 15 tokens. On the other hand, the description length appears more normally distributed, with a peak at around 30 tokens. There are 149 industries in the dataset, covering a broad range of sectors. We count the number of unique companies for each industry and present the result in Figure 11. As we can see, most industries have between 10^2 (100) and $10^{3.5}$ (3,162) companies.

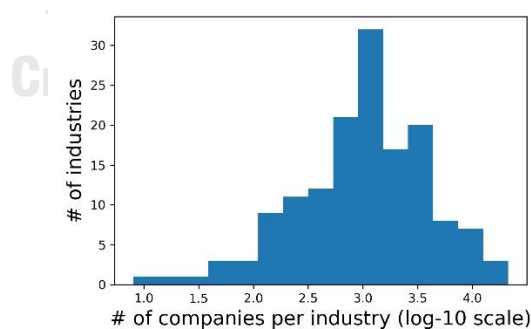


Figure 11: Distribution of the number of companies in each industry (in log-10 scale). X-axis is the number of companies belonging to an industry in log-10 scale. Y-axis is the number of industries in each bucket.

We perform some further investigation to better understand the nature of the dataset:

1. What percentage of the slogans are contained in the description? These slogans can be “generated” using a purely extractive approach.
2. What percentage of the slogan words occur in the description?
3. What percentage of the descriptions contain a company name?
4. What percentage of descriptions and slogans contain entities? What type of entity?
5. What percentage of slogans contain entities not mentioned in the description?

Entity type	Valid dataset			Test dataset		
	Desc	Slogan	Slog - Desc	Desc	Slogan	Slog - Desc
ORGANIZATION	65.6	31.3	27.8	63.9	30.2	29.7
GPE	36.7	19.4	7.5	33.5	20.2	7.5
DATE	16.4	1.3	1.0	18.2	1.9	1.4
CARDINAL	10.2	1.4	0.8	10.4	1.1	0.6
LOCATION	4.6	1.1	0.6	4.6	1.1	0.7
PERSON	4.2	2.5	1.6	3.3	1.3	0.9
PRODUCT	4.2	0.2	0.1	4.2	0.4	0.4
NORP	2.6	0.9	0.4	3.8	0.6	0.1
FACILITY	2.5	0.5	0.4	2.9	0.1	0.1
TIME	2.0	0.02	–	1.4	–	–
WORK OF ART	1.5	0.4	0.3	1.7	0.6	0.5
PERCENT	1.3	0.09	0.09	1.9	–	–
ORDINAL	1.3	0.2	0.1	1.4	0.3	0.2
MONEY	0.7	0.2	0.1	0.8	–	–
QUANTITY	0.5	–	–	0.4	0.1	–
EVENT	0.5	0.2	0.2	0.3	0.8	0.8
LAW	0.3	–	–	0.3	–	–
LANGUAGE	0.3	0.09	0.02	0.2	0.1	–

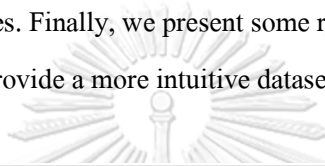
Table 14: The percentage of descriptions and slogans containing each type of entity. “Slog-Desc” refers to the percentage of entities only occur in the slogan.

First, only 11% of the slogans are contained in the corresponding input description, showing that around 90% of the cases require various degrees of abstraction instead of copying from the input. On the other hand, 62.7% and 59% of the validation and test slogan words can be found in

the corresponding descriptions. This word overlap is essential for a seq2seq model to learn the “mapping” between input and output sequences.

63.1% and 66.6% of the validation and test descriptions contain the company name. Company names are often rare words. If not dealt with, they might cause problems to the model.

To answer the fourth question, we use Stanza (Qi et al., 2020) to perform named-entity tagging on both descriptions and slogans. Table 14 shows the frequency of each type of entity. We observe that around 50% of entities occur only in the slogan but not in the corresponding description. Training a seq2seq model using the original data will likely encourage the model to hallucinate unsupported entities. Finally, we present some randomly sampled data from the validation set in Table 15 to provide a more intuitive dataset overview.



Remark	Slogan	Description
Slogan in desc	Total Rewards Software	Market Total Rewards to Employees and Candidates with Total Rewards Software . We provide engaging Total Compensation Statements and Candidate Recruitment.
100% unigrams in desc	Algebra Problem Solver	Free math problem solver answers your algebra homework questions with step-by-step explanations.
57% unigrams in desc	Most Powerful Lead Generation Software for Marketers	Powerful lead generation software that converts abandoning visitors into subscribers with our dynamic marketing tools and Exit Intent technology.
33% unigrams in desc	Business Process Automation	We help companies become more efficient by automating processes, impact business outcomes with actionable insights & capitalize on growth with smart applications.
0% unigrams in desc	Build World-Class Recreation Programs	Easily deliver personalised activities that enrich the lives of residents in older adult communities. Save time and increase satisfaction.
Entities in desc	Digital Agency in Auckland _[GPE] & Wellington _[GPE] , New Zealand	Catch Design is an independent digital agency in Auckland and Wellington , NZ. We solve business problems through the fusion of design thinking, creativity, innovation and technology.
Entities not in desc	Leading Corporate Advisory Services Provider in Singapore _[GPE] & HongKong _[GPE]	Offers Compliance Advisory services for Public listed companies, Private companies, NGOs, Offshore companies and Limited Liability Partnerships (LLPs).

Table 15: Sample (description, slogan) pairs from the validation set. We highlight the exact match words in bold.

4.3 Proposed Method

4.3.1 Model

We apply a Transformer seq2seq model to generate slogans conditioned on the input description. We choose to use BART (Lewis et al., 2020) because it benefits from a bi-directional

encoder and an auto-regressive decoder and is particularly competitive in conditioned generation tasks.

We use the pre-trained DistilBART¹² checkpoint with 6 layers in both encoder and decoder. The model is a distilled version of the original BART-large model and is roughly half the size with 230M parameters. We use the relatively small model because it requires less RAM to deploy to production and its inference time is also faster.

4.3.2 Generating truthful slogans with masking

As discussed in section 4.2, some slogans contain entities not present in the description. However, it does not necessarily mean the slogans are untruthful. E.g., consider the hypothetical (description, slogan) pair (“Knorex is a digital advertising company powered by machine learning.” and “Best digital advertising solution Singapore”), the entity “Singapore” does not appear in the company description. However, since the companies write the slogans, they are regarded as truthful. On the other hand, if a machine learning model inserts an entity not present in the input sequence, it is more likely to be hallucinated. Therefore, we apply two simple pre-processing techniques to encourage the model to generate slogans that are more grounded by the input: company name and entity masking.

Section 4.2 shows that over 60% of the company descriptions contain company names. On the other hand, company names often contain rare words or are ambiguous (due to creativity). Therefore, we hypothesize that delexicalizing the company name and replacing it with a single [MASK] token helps the model better focus on the critical information.

```

Input: company_name, text, MASK_TOKEN
Result: delexicalised_text, surface_form
delexicalised_text = text;
surface_form = company_name + “ ”;
while surface_form.contains(“ ”) do
    surface_form = surface_form.substring(0, surface_form.lastIndexOf(“ ”));
    if text.contains(surface_form) then
        delexicalised_text = text.replace(surface_form, MASK_TOKEN);
        break;
    end
end

```

Algorithm 2: Prefix matching algorithm to delexicalize company names.

¹² <https://huggingface.co/sshleifer/distilbart-cnn-6-6>

The company names are present in the Kaggle company dataset. However, they often do not occur in the identical form in the descriptions. For example, “Google LLC” are exclusively referred to as only “Google” and “Prudential Assurance Company Singapore (Pte) Limited” are referred to as “Prudential.” Motivated by this observation, we apply a prefix matching algorithm to delexicalize the company name, as shown in Algorithm 2. Table 16 provides an example of the delexicalization process for “Atlassian Corporation Plc.”

Company:	Atlassian Corporation Plc
Description:	Millions of users globally rely on Atlassian products every day for improving software development, project management, collaboration and code quality.
Surface Form:	Atlassian
Delexicalised Description:	Millions of users globally rely on <company> products every day for improving software development, project management, collaboration and code quality.

Table 16: Example description before and after company name delexicalization.

In the abstractive summarization literature, introducing irrelevant entities is referred to as *entity hallucination* (Nan et al., 2021). Based on a recent study (Gabriel et al., 2021), entity hallucination is the most common type of factual error introduced by a modern neural seq2seq model. We apply a similar method to mask entities appearing in the company description and replace them with unique identifiers.

We first tag all the named entities in the (description, slogan) pairs using Stanza (Qi et al., 2020). We focus on frequent entity types¹³: GPE, DATE, CARDINAL, LOCATION, PERSON, NORP. Note that we omit the ORGANIZATION entity type because many nominal mentions are falsely tagged as ORGANIZATION, likely due to the title case in the text.

We assign a unique identifier to each entity within a (description, slogan) pair. Therefore, if the same entity occurs in the description and the slogan, it will be assigned the same entity ID. Specifically, we maintain a counter for each type of entity. The first mention of an entity type is assigned the ID [entity_type] while subsequent mentions are assigned the ID [entity_type count]. Instead of using the upper-case acronym, we use lower-cased words with the following mapping: {GPE: country, DATE: date, CARDINAL: number, LOCATION: location, PERSON: person, NORP: national}. Table 17 exemplifies the entity masking process.

¹³ The entity type description can be found in: <https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf>.

Description:	PR-Living Belgium family-owned furniture brand with production facilities in Waregem where it brings the best of Belgian -inspired Design Upholstery & Furniture pieces to the global consumers.
Slogan:	A Belgian furniture brand
Entities:	'Belgium': GPE, 'Waregem': GPE, 'Belgian': NORP
Masked Description:	PR-Living [country] family-owned furniture brand with production facilities in [country1] where it brings the best of [national]-inspired Design Upholstery & Furniture pieces to the global consumers.
Masked Slogan:	A [national] furniture brand
Reverse Mapping:	[country]: 'Belgium', [country1]: 'Waregem', [national]: 'Belgian'

Table 17: Applying entity masking to an example description and slogan pair.

As shown in Table 14, a substantial portion of entities in the slogan does not occur in the description. Therefore, we discard all the data where *any* entity in the slogan does not appear in the description. It accounts for 10% of the data. Although this procedure reduces the training data size, it encourages the model to generate only entities present in the input sequence. For both company name delexicalization and entity masking, we perform a dictionary look-up to replace the generated mask tokens with the original company name or entity mention.

4.3.3 Generating diverse slogans with syntactic control

Generating a large variety of slogans is an effective method to combat *ad fatigue*. Throughout the lifetime of an ad campaign, if we notice a drop in users' interest in a slogan, we can replace it with a new one. While our initial system often generated plausible slogans, the slogans differ from each other only slightly. Moreover, the output are often simple noun phrases. We apply part-of-speech (POS) tagging on the slogans in our training dataset to investigate this problem. Table 18 lists the most frequent POS patterns. Nine out of ten top POS tag sequences are noun phrases, explaining why the model often generates a noun phrase. This finding motivates us to control the syntactic structure of the generation explicitly.

Inspired by CTRL (Keskar et al., 2019), we model slogan generation as a conditional generation $P(\text{slogan} \mid \text{description}, \text{ctrl})$ instead of the standard probability $P(\text{slogan} \mid \text{description})$. We use the coarse-grained POS tag of the first word in the slogan as the control code. Furthermore, we merge all adjectives and adverbs and group all infrequent POS tags as the "OTHER" category. Table 19 presents the complete list of syntactic control codes and the corresponding frequency.

POS tag sequence	Frequency	Example
NNP NNP NNP	12,892	Emergency Lighting Equipment
NNP NNP NNP NNP	5982	Personal Injury Lawyers Melbourne
NNP NNP NNPS	4217	Rugged Computing Solutions
JJ NN NN	3109	Flexible Office Space
NNP NNP NN	2789	Bluetooth Access Control
NNP NNP NNS	2632	Manchester Law Firms
NNP NNP NNP CC NNP NNP	2190	Local Programmatic Advertising & DSP Platform
NNP NNP CC NNP NNP NNP	2157	Retro Candy & Soda Pop Store
NN NN NN	2144	Footwear design consultancy
NNP NNP NNP NNP NNP	1662	Commercial Construction Company New England

Table 18: The top 10 slogan POS tag sequences in the training data with example.

Code	Frequency	Meaning
NN	208,061	All types of nouns
JJ	44,926	All types of adjectives and adverbs
VB	37,331	Verbs of any form or tense
DT	17,645	Determiners
PR	8484	Personal or possessive pronouns
OTHER	7644	Any other tags not included above, such as numbers, prepositions and question words

Table 19: Full list of syntactic control codes.

We prepend the control code to the input description with a `</s>` token as delimitator. We use the control code derived from the target slogan during training. During inference, we randomly sample control codes to prepend to the input description to generate syntactically-diverse slogans. Our method differs from CTRL in that 1) CTRL utilizes control codes during pre-training while our method uses them only during fine-tuning; 2) we use a BART encoder-decoder model while CTRL uses an autoregressive model.

Munigala et al. (2018) applied a similar technique as ours to enforce the first word of the generated sentence to be a verb. In contrast, our conditional training approach allows the model to generate text starting with different POS tags. Moreover, we let the model learn the association between words and POS tags instead of relying on hard constraints. It provides more flexibility to the model to occasionally ignore the control codes to ensure the generation is grammatical.

4.4 Experiments

Section 4.4.1 conducts a quantitative evaluation of our model against various baselines and reports the ROUGE -1/-2/-L scores on both the validation and the manually-curated test set. Next, we separately evaluate the truthfulness and diversity aspects in Section 4.4.2 and 4.4.3. Finally, we conduct a human evaluation and report the results in Section 4.4.4.

We ran all the experiments on a cloud instance with an Nvidia Quadro P5000 GPU (16 GB vRAM). We use the Hugging Face (Wolf et al., 2020)’s DistilBART implementation with a training batch size of 64 and a cosine decay learning rate (maximum learning rate=1e-4). During inference, we rely on greedy decoding so that the diversity is due to the model and not the sampling.

4.4.1 Quantitative evaluation

We compare our proposed method with the following baselines:

- *First sentence*: predicting the first sentence in the description as the slogan. Despite its simplicity, the first sentence baseline yields competitive performance for document summarization (Katragadda et al., 2009).
- *First-k words*: predicting the first k words in the description as the slogan. Slogans are often concise and shorter than a typical sentence. Therefore, we report the result of this baseline, which outputs a sequence of the expected slogan length.
- *Skeleton-based* (Tomašić et al., 2014): a skeleton-based system relying on genetic algorithms to score the slogans in each generation. We follow Tomašić et al. (2014)’s implementation except for omitting the frequent grammatical relations database and the Corpus of Contemporary American English because these resources are not available.
- *Encoder-decoder* (Bahdanau et al. 2015): a strong GRU-based encoder-decoder baseline. We follow the exact hyper-parameters as Misawa et al. (2020). Equivalent to Misawa et al. (2020) without copy mechanism and reconstruction loss.
- *Pointer-Generator* (See et al. 2017): encoder-decoder model with copy mechanism. A strong baseline for abstractive summarization. Equivalent to Misawa et al. (2020) without reconstruction loss.
- Misawa et al. (2020): A GRU model with copy mechanism to handle unknown words and reconstruction loss to encourage the model to generate distinct slogans.

Table 20 summarizes various models' performance. The first- k word baseline performed reasonably well in ROUGE scores, mainly due to the word overlap between descriptions and slogans. Figure 12 studies the impact of various k on the ROUGE score. We can observe that the model achieves the optimal ROUGE scores when k is in the range (9, 12). It also explains why the first- k word baseline outperformed the first sentence baseline, which often outputs longer sequences.

	Valid Dataset			Test Dataset		
	R1	R2	RL	R1	R2	RL
First sentence	26.12	13.03	23.88	25.50	12.73	23.47
First- k words ($k = 11$)	27.50	13.72	25.33	25.76	12.68	24.02
Skeleton-based	16.09	1.62	14.01	16.61	1.79	14.72
Encoder-Decoder	24.85	9.38	24.01	23.91	9.31	23.28
Pointer-Generator	26.42	10.15	25.63	26.24	10.67	25.65
Misawa et al. (2020)	24.14	9.19	23.37	26.01	10.00	25.39
DistilBART	36.74	18.87	33.97	34.95	17.38	32.47
DistilBART+delex	37.37	19.51	34.69	35.06	17.79	32.52
DistilBART+delex+ent	37.76	19.69	35.17	35.58	18.47	33.32

Table 20: The ROUGE -1/-2/-L F1 scores of various models on the validation and test datasets.

The skeleton-based approach performed the worst among all baselines. Although it copies salient keywords from the input sequence, injecting them into slogan skeletons often results in non-grammatical and nonsensical results.

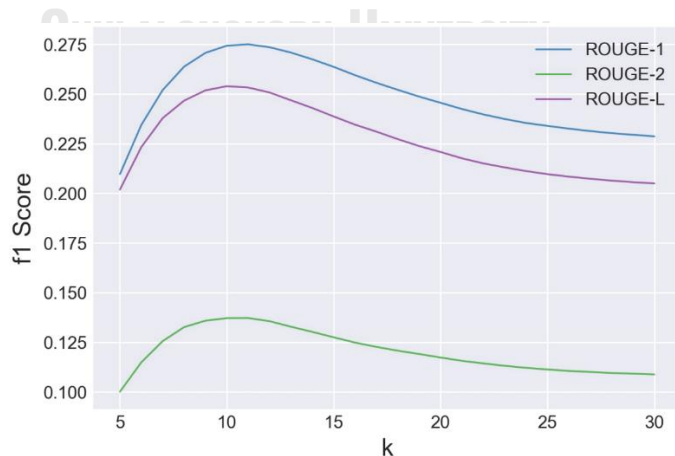



Figure 12: The first- k words baseline's ROUGE scores by varying k .

The comparison among the three GRU baselines revealed that the copy mechanism consistently improved performance. However, the contribution of reconstruction loss remains unclear. Overall the Pointer-Generator model’s ROUGE-1 and ROUGE-L scores are slightly better than the first- k words baseline but pale when compared with any BART model.

Comparing the BART models, we can see that both company name delexicalization and entity masking improved the performance. The final model achieved a ROUGE -1/-2/-L score of 35.58/18.47/33.32 on the test dataset, outperforming the best non-Transformer baseline by an absolute 10%. Table 21 shows randomly sampled example generations from various systems.



Gold:	Fast, Fresh & Tasty Mexican Food
First-k words:	We may not be the only burrito in town, but we've
Skeleton-based:	Bar to Your Burrito in Town
Pointer-Generator:	The World 's First Class Action Club
DistilBART:	The Best Burrito in Town
Gold:	A better UK energy supplier
First-k words:	Welcome to Powershop, a better gas and energy supplier. We offer
Skeleton-based:	Your Gas in Competitive Electricity, Energy Deals, Offer and Website
Pointer-Generator:	Gas and Electricity Supplier
DistilBART:	Gas and Energy Supplier
Gold:	Saigon Food Tours and City Tours Led by Women
First-k words:	Top-ranked Ho Chi Minh City food tours from the first company
Skeleton-based:	Food Company & Culture Tours Female
Pointer-Generator:	Food Tours & Food Tours
DistilBART:	Food Tours in Vietnam
Gold:	Top Engineering Colleges In Tamilnadu
First-k words:	top Engineering colleges in Tamil Nadu based on 2020 ranking. Get
Skeleton-based:	We Info Colleges!
Pointer-Generator:	Engineering College in Dehradun Uttarakhand
DistilBART:	Top Engineering Colleges in Tamil Nadu

Table 21: Sample generated slogans from different models. "Gold" is the original slogan. The DistilBART model uses both company name delexicalization and entity masking.

While the first- k words baseline occasionally has a considerable word overlap with the gold slogans, its output looks like a typical sentence rather than a slogan. Pointer-Generator sometimes generates similar slogans as DistilBART, but it is more prone to hallucinate information and generate ungrammatical content or repetitions.

4.4.2 Truthful evaluation

We apply automatic truthful evaluation methods to evaluate our model. We first rely on an entailment model following Maynez et al. (2020). Specifically, we use a ROBERTA-LARGE (Liu et al., 2019) checkpoint fine-tuned on the Multi-Genre NLI dataset (Williams et al., 2018)¹⁴ and use the entailment probability between the input description and the generated slogan to measure truthfulness.

Secondly, we use a pre-trained FactCC (Kryscinski et al., 2020) model. FactCC was trained on documents with various synthesized factual errors to measure the factual consistency between generated summaries and source documents. FactCC correlates well with human judgment on truthfulness based on Pagnoni et al. (2021)’s recent study. It was also frequently used as a truthfulness evaluation metric in subsequent works (Cao et al., 2020; Dong et al., 2020). We measure truthfulness using the predicted probability FactCC assigned to the category “consistent.”

Table 22 presents the automatic truthfulness metric scores for the baseline DistilBART model and our proposed method with delexicalization and entity masking. We can see that our performed method outperformed the baseline with a strong statistical significance.

	Valid dataset		Test dataset	
	Entailment	FactCC	Entailment	FactCC
DistilBART	75.89	70.23	70.87	71.21
DistilBART+delex+ent	83.25 (2.3e-63)	73.09 (5.1e-6)	81.61 (1.0e-21)	75.71 (8.4e-4)

Table 22: The automatic truthfulness evaluation scores of the baseline DistilBART model and our proposed method. The p -value of a double-sided paired t -test is presented in brackets.

Compared to Table 20, our method’s improvement is more pronounced. We hypothesize that it is because n -gram metrics like the ROUGE score are not very sensitive to factual errors, which often occur within a local context. For example, if the reference slogan is “Relocation Service in Barcelona,” and the model predicts “Relocation Service in Belgrade,” it will nevertheless receive a moderately high ROUGE score. However, entailment and factuality models will detect the factual inconsistency and assign a low score.

¹⁴ <https://huggingface.co/roberta-large-mnli>

4.4.3 Diversity evaluation

Because we use control codes to improve the model’s generation diversity, we first evaluate the control accuracy. We prepend each of the six control codes to all data in the test set and generate slogans. We then measure the control accuracy as the percentage of cases where the first word’s POS tag agrees with the specified control code. The result is presented in Table 23.

	Ctrl accuracy						Diversity	Abstractive
	NN	JJ	VB	DT	PR	OTHER		
W/O upsampling	92.56	37.12	61.47	93.96	97.28	90.64	46.69	45.04
Upsampling	91.14	42.35	48.69	71.83	96.88	55.53	44.81	43.85
Nucleus sampling	70.32	11.27	7.85	5.23	0.80	1.31	27.97	27.01

Table 23: Different method’s syntactic control accuracy and word diversity. We highlight the best scores in bold. All models do not use delexicalization or entity masking.

Table 19 shows that the control codes in our training data are very skewed. For example, the most frequent code, “NN”, has 208k examples, while the least frequent code “OTHER” has fewer than 8k examples. Therefore, we experimented with upsampling all control codes other than “NN” to 100k to make the data more label-balanced. We report the results of the model trained using the upsampled data in the second row of Table 23.

We compare with nucleus sampling (Holtzman et al., 2019) baseline with top- $p=0.95$, which matches human perplexity based on their paper. We note that nucleus sampling’s control accuracy should be treated as a random baseline because the model does not take the control code as input.

We calculate distinct-1 score (Li et al., 2016) to measure diversity, which is the total number of unique words in a set of slogans generated from the same input description divided by the total number of generated words.

The result indicates that our model achieved over 90% control accuracy for all control codes except for “JJ” and “VB”. The strong syntactic control accuracy shows that the model learned to associate words with their corresponding POS tags, although we did not explicitly provide the POS tag information. It suggests pretrained language models can capture linguistic information internally, as discussed in depth in Tenney et al. (2019) and Rogers et al. (2020).

Upsampling yielded worse control accuracy and diversity, likely because the model overfits repeated training examples. Compared to our proposed method, nucleus sampling also has a much lower diversity.

Furthermore, we measure abstractiveness as the percentage of words in the generated slogans that are not present in the input description. Similarly, our model is more abstractive than the baseline.

Finally, we invited an annotator to assess the generated slogan’s quality manually. We randomly sampled 50 companies from the test set and generated slogans with each of the six control codes, resulting in 300 slogans. We also sampled 300 slogans using nucleus sampling and invited the annotator to conduct a pair-wise evaluation comparing our method with nucleus sampling. We randomized the order, so the annotator was unaware of which system generated the slogans. We summarize the pair-wise comparison result in Table 24.

Code	Better	Can't decide	Worse	<i>p</i> -Value
NN	28	3	19	0.189
JJ	32	3	15	0.013
VB	36	0	14	1.86e-03
DT	41	1	8	2.43e-06
PR	37	0	13	6.89e-04
OTHER	39	0	11	7.50e-05
Overall	213	7	80	7.85e-15

Table 24: Pair-wise human evaluation result of each control code compared with the nucleus sampling baseline. We calculate the *p*-value using double-sided Wilcoxon signed-rank test.

“Better” indicates that our method generates better slogans than the baseline and vice versa for “worse”.

Except for “NN”, all control codes outperformed the nucleus sampling baseline with statistical significance ($p=0.05$). The result is expected because “NN” is the most common code in the dataset. Using “NN” as the control code will likely yield similar results to sampling without a control code. We present randomly sampled generations with different control codes in Table 25. We can see that the POS tag of the first word does not always match the specified control code. Nevertheless, the generations are diverse in both syntax and content.

4.4.4 Human evaluation

Based on previous evaluation results, we include company name delexicalization, entity masking, and random sampling from the control code set {JJ, VB, DT, PR, OTHER} into our final model. We conduct a human evaluation by letting our model and various baselines generate slogans from the same set of 50 company descriptions. We ask two human evaluators to rate each slogan in three aspects: coherence, well-formedness, and catchiness. Each aspect is rated on a scale of 1-3 (poor, acceptable, good). We average the scores assigned by the two annotators and present the result in Table 26.



Desc:	Helping eCommerce business growing their sales & revenues. Specialist in product feeds, shopping ads, conversion optimisation, SEO and website personalisation.
NN:	eCommerce Business Growth & Revenue Optimization Experts
JJ:	Ecommerce Marketing Agency in London & Essex
VB:	Helping eCommerce Business Grow Their Sales & Revenues
DT:	The eCommerce Experts
PR:	Your eCommerce Partner for Growth & Success!
OTHER:	How to Grow Your Business with eCommerce
Desc:	We are experts in: Web/Mobile/Desktop apps Development. Innovative technologies.
NN:	Web and Mobile App Development Company in India
JJ:	Innovative Technologies. Web and Mobile Apps Development Company
VB:	Leading Mobile App Development Company in India
DT:	Achieving Digital Transformation in the Cloud with Mobile Apps Development
PR:	We are experts in mobile apps development
OTHER:	Where technology meets creativity

Table 25: Randomly sampled slogans generated with different control codes.

The first sentence baseline received poor well-formedness and catchiness scores. As discussed previously, this baseline failed to generate outputs matching the slogan style. The skeleton-based approach improved catchiness over the first sentence baseline. However, it is prone to generating non-grammatical and nonsensical outputs, thus receiving the lowest well-formedness score among all baselines.

System	Coherent	Well-formed	Catchy
First sentence	3.00**	1.49**	1.19**
Skeleton-based	2.31**	1.36**	1.41**
Pointer-Generator	2.63**	2.07**	1.51**
DistilBART	2.89	2.81	1.87**
Ours	2.91	2.79	2.22

Table 26: Human evaluation on coherence, well-formedness, and catchiness. We highlight the best score for each aspect in bold (excluding the “coherent” aspect for the first sentence baseline because it is coherent by definition). ** indicates statistical significance when compared with our method using a two-sided paired *t*-test using p -value=0.005.

The Pointer-Generator baseline outperformed the previous two baselines substantially across all aspects, demonstrating the strength of modern encoder-decoder models. DistilBART further improved over Pointer-Generator, the difference in well-formedness being most pronounced. We conjecture that it is because DistilBART was pre-trained on a large corpus while Pointer-Generator was trained from scratch. Thus, DistilBART inherently has a stronger language generation capability.

Our proposed method received a similar coherence and well-formedness score as DistilBART. However, it outperformed DistilBART by a large margin in the catchiness aspect. It demonstrates that we improve the catchiness of generated slogans as a by-product by explicitly varying the syntactic structure.

4.5 Conclusions

This section proposed a novel method to generate slogans from a short description using a seq2seq Transformer. We further improved the cohesiveness and diversity of generated slogans using company name delexicalization, entity masking, and conditional training with syntactic control codes. Our final model outperformed various baselines not only in ROUGE scores but automatic and human evaluation on various aspects.

5. Conclusions

This thesis studies the application of natural language processing (NLP) techniques on advertising. While NLP has advanced dramatically in recent years with the massive success in transfer learning, there was no systematic study of how NLP can be applied to digital advertising.

Just as communication consists of two indispensable components, understanding and expressing, NLP can also be divided into two subfields: natural language understanding (NLU) and natural language generation (NLG). We study how each subfield is directly relevant to digital advertising.

For the task of understanding, we built weakly-supervised text classifiers from keywords or even the category names alone. The classifiers categorize millions of web pages per day to assign them into either a pre-defined taxonomy or custom categories. We also introduced a novel technique to evaluate the weakly-supervised classifiers without any labeled validation dataset, making the models more robust in real-world scenarios.

For the generation task, we proposed a Transformer sequence-to-sequence model to generate ad slogans from a brief description. We also introduced novel techniques to improve the truthfulness and diversity of the generation. Our final model yields statistically more catchy slogans than a previous state-of-the-art model based on human judgment.

We believe our work is only a tip of the iceberg. There is enormous potential to apply NLP techniques to digital advertising, which can benefit both advertisers and users to improve ad relevancy and ensure the appropriate ad message. We hope there will be more collaboration between ad tech companies and the research community to keep pushing this frontier.

REFERENCES



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

- Abrams, Z. and Vee, E. 2007. Personalized ad delivery when ads fatigue: An approximation algorithm. In International Workshop on Web and Internet Economics (pp. 535-540). Springer, Berlin, Heidelberg.
- Alnajjar, K. and Toivonen, H. 2021. Computational generation of slogans. *Natural Language Engineering*, 27(5), pp.575-607. Cambridge University Press.
- Bahdanau, D., Cho, K., and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA.
- Breve, F.A., Zhao, L. and Quiles, M.G. 2010. Semi-supervised learning from imperfect data through particle cooperation and competition. In The 2010 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
- Brodley, C.E. and Friedl, M.A. 1996. Identifying and eliminating mislabeled training instances. In Proceedings of the National Conference on Artificial Intelligence (pp. 799-805).
- Burton, S. and Lichtenstein, D.R. 1988. The effect of ad claims and ad context on attitude toward the advertisement. *Journal of Advertising*, 17(1), pp.3-11.
- Cao, M., Dong, Y., Wu, J. and Cheung, J.C.K. 2020. Factual Error Correction for Abstractive Summarization Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 6251-6258).
- Carbonell, J. and Goldstein, J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 335-336).
- Chang, M.W., Ratnoff, L.A., Roth, D., and Srikumar, V. 2008. Importance of semantic representation: Dataless classification. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, Chicago, Illinois, Volume 2. Association for the Advancement of Artificial Intelligence, pp. 830–835.
- Charoenphakdee, N., Lee, J., Jin, Y., Wanvarie, D., and Sugiyama, M. 2019. Learning only from relevant keywords and unlabeled documents. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China. Association for Computational Linguistics, pp. 3984–3993.

- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, Minneapolis, Minnesota.
- Dietterich, T.G., 2000. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2), pp.139-157.
- Dong, Y., Wang, S., Gan, Z., Cheng, Y., Cheung, J.C.K. and Liu, J. 2020. Multi-Fact Correction in Abstractive Text Summarization. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 9320-9331).
- Donmez, P., Lebanon, G. and Balasubramanian, K. 2010. Unsupervised supervised learning I: estimating classification and regression errors without labels. *Journal of Machine Learning Research*, 11(4).
- Druck, G., Mann, G., and McCallum, A. 2008. Learning from labeled features using generalized expectation criteria. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Singapore, Singapore. Association for Computing Machinery, pp. 595–602.
- Elhadad, N. and Sutaria, K. 2007. Mining a lexicon of technical terms and lay equivalents. In Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, pp. 49-56.
- Frénay, B. and Verleysen, M. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5), pp.845-869.
- Gabriel S., Celikyilmaz A., Jha R., Choi Y. and Gao J. 2021. GO FIGURE: A meta evaluation of factuality in summarization. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online. Association for Computational Linguistics, pp. 478–487.
- Gabrilovich, E., Markovitch, S. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, Hyderabad, Volume 7. International Joint Conferences on Artificial Intelligence, pp. 1606–1611.

- Gamberger, D., Lavrac, N. and Groselj, C. 1999. Experiments with Noise Filtering in a Medical Domain. In Proceedings of the Sixteenth International Conference on Machine Learning (pp. 143-151).
- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M. and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1), pp.2096-2030.
- Gatti, L., Özbal, G., Guerini, M., Stock, O., and Strapparava, C. 2015. Slogans are not forever: Adapting linguistic expressions to the news. In Proceedings of the 24th International Joint Conference on Artificial Intelligence, pp. 2452–2458, Buenos Aires, Argentina.
- Gatti, L., Özbal, G., Stock, O., and Strapparava, C. 2017. To sing like a mockingbird. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 298–304, Valencia, Spain.
- Gerlach, R. and Stamey, J. 2007. Bayesian model selection for logistic regression with misclassified outcomes. *Statistical Modelling*, 7(3), pp.255-273.
- Hai, Z., Chang, K. and Cong, G. 2012. One seed to find them all: mining opinion features via association. In Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 255-264.
- Holtzman, A., Buys, J., Du, L., Forbes, M. and Choi, Y. 2019. The Curious Case of Neural Text Degeneration. In International Conference on Learning Representations. New Orleans, Louisiana.
- Howard, J., and Ruder, S. 2018. Universal language model fine-tuning for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers, pp. 328-339, Melbourne, Australia.
- Hu, M. and Liu, B. 2004. Mining opinion features in customer reviews. In Proceedings of the 19th National Conference on Artificial Intelligence, pp. 755-760.
- Hughes, J.W., Chang, K.H. and Zhang, R. 2019. Generating better search engine text advertisements with deep reinforcement learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2269-2277).
- Iwama, K., and Kano, Y. 2018. Japanese advertising slogan generator using case frame and word vector. In Proceedings of the 11th International Conference on Natural Language Generation, pp. 197–198, Tilburg, The Netherlands.

- Jaffe, A., Nadler, B. and Kluger, Y. 2015. Estimating the accuracies of multiple classifiers without labeled data. In *Artificial Intelligence and Statistics*, pp. 407-415. PMLR.
- Jin, Y., Wanvarie, D., and Le, P. 2017. Combining lightly-supervised text classification models for accurate contextual advertising. In *Proceedings of the 8th International Joint Conference on Natural Language Processing, Volume 1: Long Papers*, Taipei, Taiwan.
- Jin, Y., Wanvarie, D., and Le, P. 2022. Learning from noisy out-of-domain corpus using dataless classification. *Natural Language Engineering*, 28(1), pp.39-69. Cambridge University Press.
- Jin, Y., Bhatia, A. and Wanvarie, D. 2021a. Seed Word Selection for Weakly-Supervised Text Classification with Unsupervised Error Estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop* (pp. 112-118).
- Jin, Y., Kadam, V. and Wanvarie, D. 2021b. Bootstrapping Large-Scale Fine-Grained Contextual Advertising Classifier from Wikipedia. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)* (pp. 1-9).
- Jin, Y., Bhatia, A., and Wanvarie, D. In press. Toward Improving Coherence and Diversity of Slogan Generation. *Natural Language Engineering*. Cambridge University Press.
- Kanungo, Y.S., Negi, S. and Rajan, A. 2021. Ad Headline Generation using Self-Critical Masked Language Model. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers* (pp. 263-271).
- Katragadda, R., Pingali, P. and Varma, V. 2009. Sentence Position revisited: A robust light-weight Update Summarization ‘baseline’ Algorithm. In *Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies (CLIAWS3)* (pp. 46-52).
- Keskar, N. S., McCann, B., Varshney, L., Xiong, C., and Socher, R. 2019. CTRL – A conditional transformer language model for controllable generation. ArXiv preprint arXiv:1909.05858.
- King, B., and Abney, S.P. 2013. Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 1110–1119, Atlanta, USA.

- Krizhevsky, A., Sutskever, I. and Hinton, G.E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, pp.1097-1105.
- Kryscinski, W., McCann, B., Xiong, C. and Socher, R. 2020. Evaluating the Factual Consistency of Abstractive Text Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 9332-9346).
- Lang, K. 1995. NewsWeeder: learning to filter netnews. In *Proceedings of the Twelfth International Conference on International Conference on Machine Learning* (pp. 331-339).
- Langheinrich, M., Nakamura, A., Abe, N., Kamba, T. and Koseki, Y. 1999. Unintrusive customization techniques for web advertising. *Computer Networks*, 31(11-16), pp.1259-1272.
- Le, Q. and Mikolov, T. 2014. Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188-1196). PMLR.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880.
- Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, CA, USA.
- Li, C., Xing, J., Sun, A., and Ma, Z. 2016. Effective document labeling with very few seed words: A topic model approach. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. Association for Computing Machinery*, pp. 85–94, Indianapolis, USA.
- Li, C., Chen, S., Xing, J., Sun, A., and Ma, Z. 2018. Seed-guided topic model for document filtering and classification. *ACM Transactions on Information Systems (TOIS)* 37(1), 9.
- Li, C., Zhou, W., Ji, F., Duan, Y., and Chen, H. 2018. A deep relevance model for zero-shot document filtering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers. Association for Computational Linguistics*, pp. 2300–2310, Melbourne, Australia.

- Li, J., Galley, M., Brockett, C., Gao, J. and Dolan, W.B. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 110-119).
- Li, X., and Yang, B. 2018. A pseudo label based dataless naive bayes algorithm for text classification with seed words. In Proceedings of the 27th International Conference on Computational Linguistics. Association for Computational Linguistics, pp. 1908–1917, Santa Fe, New Mexico, USA.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
- Lucas, D.B. 1934. The optimum length of advertising headline. *Journal of Applied Psychology*, 18(5), p.665.
- Luong, M.-T., Pham, H., and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421, Lisbon, Portugal.
- Maynez, J., Narayan, S., Bohnet, B. and McDonald, R. 2020. On Faithfulness and Factuality in Abstractive Summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 1906-1919).
- Mekala, D. and Shang, J. 2020. Contextualized weak supervision for text classification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 323-333.
- Meng, Y., Shen, J., Zhang, C., and Han, J. 2018. Weakly-supervised neural text classification. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management. Association for Computing Machinery, pp. 983–992, Turin, Italy.
- Meng, Y., Shen, J., Zhang, C. and Han, J. 2019. Weakly-supervised hierarchical text classification. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 6826-6833).

- Meng, Y., Zhang, Y., Huang, J., Xiong, C., Ji, H., Zhang, C. and Han, J. 2020. Text classification using label names only: A language model self-training approach. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 9006–9017.
- Mieder, B. and Mieder, W. 1977. Tradition and innovation: Proverbs in advertising. *Journal of Popular Culture*, 11(2), p.308.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, pp. 3111–3119, Lake Tahoe, Nevada, USA.
- Misawa, S., Miura, Y., Taniguchi, T., and Ohkuma, T. 2020. Distinctive slogan generation with reconstruction. In Proceedings of the Workshop on Natural Language Processing in E-Commerce, pp. 87–97.
- Munigala, V., Mishra, A., Tamilselvam, S. G., Khare, S., Dasgupta, R., and Sankaran, A. 2018. Persuaide! An adaptive persuasive text generation system for fashion domain. In Companion Proceedings of the The Web Conference, pp. 335–342, Lyon, France.
- Nam, J., Menca, E.L., and Fürnkranz, J. 2016. All-in text: Learning document, label, and word representations jointly. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. Association for the Advancement of Artificial Intelligence. Phoenix, Arizona, USA.
- Nam, J., Mencía, E.L., Kim, H.J. and Fürnkranz, J. 2017. Maximizing subset accuracy with recurrent neural networks in multi-label classification. In Proceedings of the 31st International Conference on Neural Information Processing Systems (pp. 5419-5429).
- Nan, F., Nallapati, R., Wang, Z., dos Santos, C., Zhu, H., Zhang, D., McKeown, K. and Xiang, B. 2021. Entity-level Factual Consistency of Abstractive Text Summarization. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (pp. 2727-2733).
- Nettleton, D.F., Orriols-Puig, A. and Fornells, A., 2010. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial intelligence review*, 33(4), pp.275-306.

- Özbal, G., Pighin, D., and Strapparava, C. 2013. Brainsup: Brainstorming support for creative sentence generation. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers, pp. 1446–1455, Sofia, Bulgaria.
- Pagnoni, A., Balachandran, V. and Tsvetkov, Y. 2021. Understanding Factuality in Abstractive Summarization with FRANK: A Benchmark for Factuality Metrics. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 4812-4829).
- Pan, S.J. and Yang, Q. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), pp.1345-1359.
- Pappas, N., and Henderson, J. 2019. GILE: A generalized input-label embedding for text classification. *Transactions of the Association for Computational Linguistics* 7, 139–155.
- Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L. 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) (pp. 2227-2237).
- Phillips, B.J. and McQuarrie, E.F. 2009. Impact of advertising metaphor on consumer belief: Delineating the contribution of comparison versus deviation factors. *Journal of Advertising*, 38(1), pp.49-62.
- Platanios, E.A., Blum, A. and Mitchell, T. 2014. Estimating accuracy from unlabeled data. In Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, pp. 682-691.
- Platanios, E. A., Dubey, A., and Mitchell, T. 2016. Estimating accuracy from unlabeled data: A bayesian approach. In Proceedings of the International Conference on Machine Learning. PMLR, pp. 1416-1425.
- Ribeiro, M.T., Singh, S. and Guestrin, C. 2016. “Why should i trust you?” Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).
- Qi, P., Zhang, Y., Zhang, Y., Bolton, J. and Manning, C.D. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (pp. 101-108).

- Rogers, A., Kovaleva, O. and Rumshisky, A. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8, pp.842-866.
- Sachan, D., Zaheer, M. and Salakhutdinov, R. 2018. Investigating the Working of Text Classifiers. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 2120-2131).
- See, A., Liu, P.J. and Manning, C.D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1073-1083).
- Settles, B. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1467–1478, Edinburgh, Scotland.
- Sun, J.W., Zhao, F.Y., Wang, C.J. and Chen, S.F. 2007. Identifying and correcting mislabeled training instances. In *Future generation communication and networking (FGCN 2007) (Vol. 1, pp. 244-250)*. IEEE.
- Sutskever, I., Vinyals, O., and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pp. 3104–3112, Montreal, Quebec, Canada.
- Tomašić, P., Znidaršič, M., and Papa, G. 2014. Implementation of a slogan generator. In *Proceedings of 5th International Conference on Computational Creativity*, Volume 301, pp. 340–343, Ljubljana, Slovenia.
- Swartz, T.B., Haitovsky, Y., Vexler, A. and Yang, T.Y. 2004. Bayesian identifiability and misclassification in multinomial data. *Canadian Journal of Statistics*, 32(3), pp.285-302.
- Tenney, I., Das, D. and Pavlick, E. 2019. BERT Rediscovered the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4593-4601).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, Long Beach, CA, USA.

- Wang, S.I. and Manning, C.D. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 90-94).
- Wang, J., Zhang, W. and Yuan, S. 2017. Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting. *Foundations and Trends® in Information Retrieval*, 11(4-5), pp.297-435.
- Yi, Y. 1991. The influence of contextual priming on advertising effects. *ACR North American Advances*.
- Wang, Z., Mekala, D. and Shang, J. 2021. X-Class: Text classification with extremely weak supervision. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 3043-3053.
- White, G.E. 1972. Creativity: The x factor in advertising theory. *Journal of Advertising*, 1(1), pp.28-32.
- Williams, A., Nangia, N. and Bowman, S. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (pp. 1112-1122).
- Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., Cistac, P., Funtowicz, M., Davison, J., Shleifer, S. and Louf, R. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 38-45).
- Yin, W., Hay, J., and Roth, D. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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). Association for Computational Linguistics, pp. 3905–3914, Hong Kong, China.
- Zhang, J., Zhao, Y., Saleh, M., and Liu, P. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the International Conference on Machine Learning. PMLR, pp. 11328–11339.

VITA

NAME Yiping Jin

DATE OF BIRTH 29 March 1990

PLACE OF BIRTH Beijing

INSTITUTIONS ATTENDED Chulalongkorn University
National University of Singapore

HOME ADDRESS Room 704, C Residence,
28 Rama 6 Soi 5,
Pathumwan, Bangkok 10330

PUBLICATION Jin, Y., Bhatia A., Wanvarie, D., and Le, P. In press. Toward improving coherence and diversity of slogan generation. *Natural Language Engineering*. Cambridge University Press.

Jin, Y., Wanvarie, D., and Le, P. 2022. Learning from noisy out-of-domain corpus using dataless classification. *Natural Language Engineering*, 28(1), pp.39-69. Cambridge University Press.

Jin, Y., Bhatia A., and Wanvarie, D. 2021. Seed word selection for weakly-supervised text classification with unsupervised error estimation. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*.

Jin, Y., Kadam V., and Wanvarie, D. 2021. Bootstrapping large-scale fine-grained contextual advertising classifier from wikipedia. *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*.

Charoenphakdee, N., Lee, J., Jin, Y., Wanvarie, D. and Sugiyama, M. 2019. Learning only from relevant keywords and unlabeled documents. 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) (pp. 3993-4002).

Jin, Y., Wanvarie, D., and Le, P. 2019. Bridging the gap between research and production with code. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 277-288, Macau, China.

