Pixie: Preference in Implicit and Explicit Comparisons

Anonymous ACL submission

Abstract

Comparisons are prevalent in text such as user reviews and are crucial for mining business intelligence. Previous studies on Comparative Preference Classification (CPC) focus on sentences where two target entities are explicitly mentioned and directly compared, such as *A is faster than B*. However, we have found that real-world comparisons are often implicit (*B is slow*) or indirect (*A is here immediately while B takes forever*). We present Pixie, a manually annotated dataset for preference classification from app reviews. Pixie contains explicit comparisons as well as sentences that implicitly reveal user preferences without linguistic comparative structures. We experiment with traditional machine learning methods with sentence embeddings and transformer-based methods on Pixie to identify preferred entities in comparative sentences. We also compare our results with the existing state-of-the-art model in CPC trained on Pixie. We find that transformer-based pretrained models fine-tuned on Pixie can achieve a weighted average F1 score of 85.23% and notably outperform the previous state-of-the-art method (73.99%).

1 Introduction

There is a Chinese saying, *huo bi san jia*, which means you are supposed to compare at least three stores before committing to a purchase. It is only natural for consumers to search for information about comparisons among competitors before purchasing a product. Consumers often rely on preferences of other consumers to make a purchasing decision. Studies show that, when making a purchasing decision, people are more likely to rely on recommendations and preferences of other users than advertisement (Nielsen, 2012; Duan et al., 2008; Chevalier and Mayzlin, 2006). Word-of-mouth marketing is the primary reason behind 20% to 50% of all purchasing decisions and is one of the most influential factors in building consumer trust in a product (Nielsen, 2020). With increased online shopping platforms and sales, the word-of-mouth conversations are migrating to online user reviews.

Online user reviews contain a cornucopia of information about user preferences and expectations about a product. Reviewers often express their opinions – likes and dislikes – on a product by comparing it against its competitors. Users who have experienced different products in the same line hold certain expectations of how these products should be, which are often expressed in comparative reviews. Such comparative reviews are helpful to other consumers in choosing a product and to companies in keeping up with the competition.

Comparative reviews can be used to rank products (Feldman et al., 2007; Kurashima et al., 2008) and extract comparative relations between products (Xu et al., 2011). Analysis of such reviews involve two tasks: (1) identifying comparative sentences from app reviews, *comparative sentence identification* (CSI)(Jindal and Liu, 2006), and (2) identifying the preferred entity in such sentences, *comparative preference classification* (CPC) (Ganapathibhotla and Liu, 2008). We focus on the second task. Identifying the preferred entity in a comparison is a non-trivial task and helps fine-grained mining of user preferences.

Prior work on identifying user preferences from text (Ganapathibhotla and Liu, 2008; Panchenko et al., 2019) focuses on explicit comparisons, i.e. sentences including mentions of two entities that are being compared. Further, strategies to extract comparative sentences from text that rely on comparative keywords (linguistic cues) (Jindal and Liu, 2006; Ganapathibhotla and Liu, 2008) or pattern matching (Li et al., 2017; Feldman et al., 2007), may overlook sentences revealing preference without any explicit comparative sentence structure.

Comparative sentences in user generated text such as reviews often lack comparative linguistic

Table 1: Examples of comparative sentences from reviews.

	Sentence	App
S_1	Bye ***Uber***, hello ***Lyft***.	*Uber*
S_2	Does ***this app*** really need to be 260 MB when the ***Marriott app*** is only 47 MB?	*Hilton Honors*
S_3	Beats the pants off ***pandora***.	*Spotify*
S_4	Great music lots of selections, plays outside of the app and a lot better then ***Spotify***.	*Amazon Music*

structures (e.g., sentence S_1 in Table 1) or may omit the entity being compared in the text (e.g., sentence S_3 in Table 1). These sentences are comparative by the virtue of expressing a preference. Such sentences are common in reviews but have been understudied by prior research.

We present Pixie (Preference in Implicit and Explicit Comparisons), a dataset for preference classification from online user reviews. We extend Jindal and Liu's (Jindal and Liu, 2006) definition of comparative sentences to include sentences that express a preference between two entities. A *comparative sentence* is a sentence containing similarity, dissimilarity, or preference between two entities. A comparison can be *implicit* (mentioning only one entity being compared, e.g., sentence S_3 in Table 1) or *explicit* (mentioning both entities being compared, e.g., sentence S_2 in Table 1).

Pixie contains 8,990 comparative sentences extracted from app reviews on Apple App Store[1] for 797 pairs of apps. Of the 8,990 comparative sentences, 4,867 contain implicit and 4,023 contain explicit comparisons. Each sentence is annotated with the preferred app, which can be the CURRENT app (the app being reviewed), OTHER app (app being compared to), or NONE (no preference).

We experiment with traditional machine learning methods with word-embeddings and transformer-based models on Pixie to identify preferred entity in a user generated comparative sentence. We also compare our results with ED-GAT (Ma et al., 2020), a state-of-the-art model for preference classification task. We find that transformer-based pretrained models fine-tuned on Pixie achieve higher F1-scores (85.23%) than the state-of-the-art (F1-score 73.99%) or traditional machine learning models (F1-score 71.86%).

Organization. The rest of this paper is organized as follows. Section 2 describes prior research on comparative preference classification (CPC) and existing datasets for this task. Section 3 introduces definitions and methodology for dataset creation and experiments. Section 4 demonstrates the results of our experiments. Section 5 contains discussion on results and merits, as well as limitations and future work to address those limitations. The paper concludes in Section 6.

2 Related Work

We discuss prior relevant research works on (1) *computational approaches* proposed by previous research to identify contrasting opinions, comparative sentences, and preference in comparisons and their applications, and (2) *existing datasets* for the task and their limitations.

2.1 Computational Approaches

Comparative sentence structures have been a subject of linguistic research producing syntactic and semantic theories on comparative clauses and predicates (Bresnan, 1973; Stechow, 1984; van Rooij, 2011). Early studies on comparative sentences in computational linguistics include syntactic and semantic handling of comparative constructions with a focus on comparative ellipsis (Rayner and Banks, 1988, 1990), comparative structures in question-answering systems (Ballard, 1988), a syntactic approach using quantifiers to identify comparisons (Friedman, 1989), and a model for semantic interpretation of comparatives based on semantic copying (Staab and Hahn, 1997).

Jindal and Liu (2006) propose a machine learning classifier based on class sequential rules with multiple minimum support to identify comparative sentences from consumer reviews, discussion forums and news articles. Ganapathibhotla and Liu (2008) propose one-sided association (OSA), a measure of association between comparative words and entity features. They handcraft rules based on comparative quantifiers and superlatives to identify preferred entity in a comparative sentence. They focus on pros and cons discussed in online product reviews to determine the directionality of comparatives in context-dependent opinions (example, *longer* is preferred for battery life but not for wait time). However, they use a list of opinion words (Hu and Liu, 2004) to identify comparative words

[1] https://apps.apple.com/us/genre/ios/id36

which limits their system to identify comparative sentences that contain an explicit opinion word. Further, computation for OSA depends on the count of words in the corpus making it highly sensitive to changes in the training dataset.

Kessler and Kuhn (2013) adopt a semantic role labelling approach to identify comparative sentences in product reviews. Their model identifies comparative predicates that contain the two entities being compared and the aspect of comparison. Panchenko et al. (2019) propose a gradient boosting model based on pretrained sentence embeddings to identify preferred entity in a comparison between two entities. Ma et al. (2020) train a novel Entity-aware Dependency based Deep Graph Attention Network (ED-GAT) on the CompSent-19 (Panchenko et al., 2019) dataset to identify preferred entity in a comparative sentence. They build a multi-hop graph attention network over a dependency graph to identify preferred entity in a comparative sentence.

Lerman and McDonald (2009) propose a contrastive summarizer using a Sentiment Aspect Match (SAM) model. Their model generates summaries highlighting the differences between two entities compared in product reviews. Wang et al. (2018b) analyze the competitive advantages and disadvantages of two competing products from online product reviews. They leverage Latent Dirichlet Allocation (LDA) to analyze differences in discussion topics across products in positive and negative reviews. Xu et al. (2011) present a graphical model to extract and visualize comparative relations between products to improve marketing and design strategies. Comparative sentences from user reviews and discussion forums have been leveraged to compare and rank similar products (Feldman et al., 2007; Kurashima et al., 2008). In other domains, comparative sentences have been studied in biomedical text to identify compared entities, aspects, and the scale of the comparison (Gupta et al., 2017), and in identifying comparative claims in scientific articles (Park and Blake, 2012).

2.2 Existing Datasets

Jindal and Liu (2006) define the *Comparative Sentence Identification* (CSI) task as identifying sentences that express a relation based on similarities or differences of more than one object. They adopt a set of keywords to create a binary classification dataset containing comparative and non-comparative sentences from consumer reviews, discussion forums, and news articles. Ganapathibhotla and Liu (2008) extend this work further to *Comparative Preference Classification* (CPC), the task of identifying preferred entity in a given comparative sentence. However, their dataset is small with a skewed distribution. The dataset contains only 837 comparative sentences, 84% of which express the preference as the first entity that appear in the sentences over the second.

Kessler and Kuhn (2014) address this gap by conducting a more fine-grained analysis of comparative sentences that reveal preference. They create a dataset for comparative preference classification task and use crowdsourcing to annotate comparative sentences by identifying comparison predicates, entities being compared, aspect of comparison, and comparison type (gradable or non gradable). However, in addition to being small and skewed (1,707 comparative sentences, 15% of the total dataset), their dataset lacks diversity as it uses reviews of a single product type (digital cameras).

Panchenko et al. (2019) create CompSent-19, a cross-domain dataset for comparative argument mining, using sentences from DepCC (Panchenko et al., 2018), a large web-scale text corpus. CompSent-19 contains 7,199 sentences annotated with preferred entity in a comparative sentence. Only 27% of those sentences contain a preference, and 70% of them prefer the entity mentioned first in the sentence.

The prior work on preference classification focuses on explicit comparative sentences (when both entities being compared are mentioned) and overlook sentences that omit the mention of one of the entities. There is a lack of study on indirect comparative sentences that lack comparative linguistic cues, such as comparative quantifiers (e.g., more than, less than) and superlative quantifiers (e.g., at least, at most, largest). In this work, we explore approaches that can identify preferred entity in implicit and indirect comparisons in addition to explicit comparisons.

3 Method

3.1 Definitions

Staab and Hahn (1997) identify three types of comparative sentences: (1) comparative sentences in which semantic and conceptual criteria formulate to complement syntactic criteria (Rayner and Banks, 1990), (2) comparative sentences in which compar-

Table 2: Examples from Staab and Hahn (1997) demonstrating *omitted complements* and *metonymies* in comparative sentences.

	Sentence
S_{oc}	The LED line of the printer 810 is clearly thicker *[than the one of the 410]*.
S_m	The Oki-X11 is faster than the *[printer manufactured by]* HP

Table 3: Types of comparative sentences.

	Sentence	Review
S_{exp1}	If ***Uber*** had customer service that could be ***Lyft***.	*Lyft*
S_{exp2}	I think that ***it's*** a lot more fun than ***temple Run***.	*Subway Surfers*
S_{imp}	For now, I'm going back to using ***Safari*** or ***Chrome***.	*Firefox*

ison relies solely on semantic or conceptual criteria with no syntactic triggering condition, and (3) comparative sentences in which comparison is intertwined with referential and textual phenomena such as *omitted complements*, and *metonymies*.

An omitted complement refers to the omission of one of the entities under comparison that can be inferred based on the context. Metonymies are a referential phenomenon when an entity in a sentence is substituted with a broader related concept representing that entity. In Table 2, sentence S_{oc} is an example of omitted complement and sentence S_m presents a metonymy (Staab and Hahn, 1997). In each case, part of the text (in brackets) is not explicitly present in the sentence but can be inferred based on the context. Pixie includes Staab and Hahn's (Staab and Hahn, 1997) second and third interpretations of comparative sentences, which have been neglected by earlier works.

Comparative sentence A comparative sentence is defined as a sentence containing similarity, dissimilarity, or a preference between two entities.

We do not rely on comparative linguistic structures or keywords (linguistic cues) in identifying a sentence as a comparative sentence. Consequently, we include *indirect comparisons*. Indirect comparative sentences are sentences that reveal a preference but lack any explicit comparative linguistic structure or cues (such as better, worse, more, and less). A comparison is *implicit* if it mentions only one of the entities being compared in the text, and *explicit* if it mentions both. Previous studies have focused on explicit comparative sentences, we focus on indirect and implicit comparisons in this work.

In Table 3, example S_{exp1} shows an explicit comparison between two apps where both apps are mentioned by name (*Uber* and *Lyft*). It is also an indirect comparison as it lacks any comparative cues, such as comparative quantifiers or superlatives. Sentence S_{exp2} shows an example of an explicit comparison when one app is mentioned by name (*temple run*) and the other is a pronominal reference (*it's* refers to *Subway surfers*, the app under review). Sentence S_{imp} shows an implicit comparison when only one of the apps being compared is mentioned. This sentence is comparative in spite of not containing a comparative structure because it reveals a preference.

Preferred entity A preferred entity is defined as the entity chosen over the other based on an explicit or implicit preference revealed in a comparative sentence.

A preferred entity can be the CURRENT app (app being reviewed), OTHER app (competitor app), or NONE (i.e. no preference). We consider cases where a preference cannot be determined (e.g., sentences S_{4p} in table 4) and where non-gradable comparatives (such as *like*, *as ...*, *as*, and *similar to*) link the entities (e.g., sentences S_{3p} in Table 4), the same as no preference. The preferred entity in a sentence can be revealed implicitly (e.g., sentence S_{2p} in Table 4) without any comparative linguistic cues or explicitly with the statement of its preference over another entity (e.g., sentences S_{1p} in Table 4).

Table 4: Examples sentences showing preference.

	Sentence	Review
S_{1p}	***This app*** compares very poorly with ***Discover's app***.	*Chase Mobile*
S_{2p}	I prefer the ***BBC app***.	*USA Today*
S_{3p}	Just as good as ***Uber*** app.	*Lyft*
S_{4p}	Makes me want to switch back to ***Pandora***, but ***it's*** just as bad.	*Spotify*

3.2 Dataset

Leveraging linguistic cues (or keywords) to identify comparative sentences yields high recall but low precision, leading to a skewed distribution where most sentences are non-comparative (Jindal and Liu, 2006). We instead adopt a list of competitor apps to identify comparative sentences from app reviews.

First, we selected 179 popular apps on Apple App Store and manually grouped them into 23 genres, including *banking* apps, *airline* apps, *weather* apps, *communication* apps, etc. Note that these genres are more fine-grained than Apple's definitions, so that apps within the same genre are direct competitors. For example, Instagram, Facebook, and Snapchat are all *communication* apps and, therefore, are competitors to each other.

Second, we collected reviews of these apps, and kept the sentences that contained mentions of their competitors, because such sentences were more likely to contain comparisons. We also included common words (or abbreviations) that could be used to refer to the apps on our list along with their full official names, e.g., *Insta* for Instagram, *FB* for Facebook, and *AA* for American Airlines.

Finally, we leveraged manual annotations to identify the preferred entity in each sentence.

Considering mentions of competitor apps instead of comparative linguistic cues to identify comparative sentences provides two major advantages. *First*, this method helps identify comparative sentences that lack explicit comparatives, (second interpretation of comparative sentences (Staab and Hahn, 1997)). For example, in sentence S_{exp1} in Table 3, it is clear that the two apps are bing compared and a preference is revealed (Lift > Uber) in spite of there being no comparative linguistic structure or cues. *Second*, the comparative class includes various interesting cases (as the context is of competitive apps in a sentence) and the resulting dataset is less skewed.

Early in the annotation process, we realized that some product names did not help in extracting comparative sentences. To improve the precision in filtering comparative sentences, we decided to de-list some product names. We removed widely used brand names such as Google, Microsoft, and Facebook as they often appeared to provide contexts as opposed to being a product. We also removed app names synonymous with or formed of common words, such as Box (cloud storage), Seamless (food delivery app), and Line (communication app). Removing these common product names increases the precision in identifying potential comparative sentences before manual labeling.

We split reviews into sentences and identified candidate comparative sentences that mention a competitor app. We annotated each sentence for *comparison* and *preferred entity*. For comparisons, we annotated each sentence as either NONCOMPARATIVE (no comparison), IMPLICIT (only one of the entities being compared mentioned in the text) or EXPLICIT (both entities being compared mentioned in the text). For preferred entity, we annotated each sentence as CURRENT (app being reviewed is preferred), OTHER (competitor app is preferred) or NONE (no preference or preference not clear).

Annotation for the dataset was conducted in three phases. In Phase 1, three of the authors annotated a sample dataset of 300 sentences based on an initial set of definitions and resolved any disagreements via discussion. This process was carried out for three iterations and produced annotation instructions for Phase 2 of labeling. In Phase 2, each author labeled an equal number of sentences and each sentence was labeled by two annotators. Disagreements between annotators were resolved by the first author. Phase 3 involved crowdsourcing with 42 annotators (graduate and undergrad students in computer science), who each labeled 400 sentences. Each sentence in this phase was labeled by three annotators and the final label was chosen based on a majority vote. When there was no clear majority, the first author acted as the tie breaker. We have obtained the Institutional Review Board (IRB) approval for this annotation task.

In Phase 2, 4,793 data instances were labeled with an inter-rater agreement (Krippendorff alpha) of 0.74 and 0.82 between each pair of annotators for the two annotation tasks, comparison type and preferred entity, respectively. In phase 3, 5,559 data points were labeled via crowdsourcing with an inter-rater agreement (Krippendorff alpha) among each triple of annotators of 0.51 and 0.74 for the two tasks, respectively. After removing some duplicative sentences, the combined annotated dataset consists of 10,287 sentences manually labeled for comparison and preferred entity.

To focus on comparative sentences, we drop the non-comparative sentences from our dataset. The final version of Pixie contains 8,890 comparative sentences for 797 competitor app pairs manually an-

notated for the type of comparison (implicit or explicit) and the preferred entity (CURRENT, OTHER, or NONE). To the best of our knowledge, Pixie is the largest dataset of comparative sentences with both explicit and implicit comparisons. Table 5 shows the distribution of labels for each class in Pixie.

Table 5: Pixie Dataset Distribution.

Preferred Entity	Comparison Type		Total
	Implicit	Explicit	
CURRENT	1,910	2,097	4,007
OTHER	2,199	1,069	3,268
NONE	758	857	1,615
Total	4,867	4,023	8,890

To ensure that the dataset can be used for training a general purpose preference identification model, we *mask* all app mentions in each sentence. Masking is a mechanism to skip certain input tokens when processing the data by encoding those tokens with some predefined tag. Without masking, the model may learn to differentiate between classes based on ground truth app preferences of users since we have limited app pairs in our dataset. We replace all entity mentions with two predefined tags, *current_app* (for app being reviewed) and *other_app* (for the competitor apps). Masking ensures that the model trained on Pixie learns the comparative and preference-revealing linguistic semantics and not just the preference between the apps being compared.

We mask all app names in sentences from Pixie using a two step semi-automated process. The first step involves automatically identifying app mentions and substituting with a predefined tag. Since we identified comparative sentences using a competitor app pair list, each sentence in Pixie contains a mention of the other (competitor) app by name. We leverage the same app pair list to identify mentions of other (competitor) apps in all the sentences automatically via keyword matching and substitute the mentions with the *other_app* tags. This step leaves us with identifying current app mentions in explicit comparative sentences, since implicit comparisons by definition only mention other apps. We employ the same programmatic approach to identify current app mentions and substitute them with the *current_app* tags. In the second step, the remaining 2,287 explicit comparative sentences that contain a pronominal reference for the current app are annotated manually by the authors. Table 6 shows example sentences masked for app mentions.

Pixie is publicly available (link removed for anonymity).

3.3 Experiments

Now, we discuss our experiments on Pixie. We experiment with traditional and deep learning (transformer based) models to identify preferred entity in a comparative sentence. We also train ED-GAT (Ma et al., 2020), the existing state-of-the-art model for the comparative preference classification (CPC) task, on Pixie and compare the performance.

3.3.1 Traditional Machine Learning Approaches

We use SBERT (SentenceBERT) (Reimers and Gurevych, 2019) to obtain sentence embeddings of each masked sentence. SBERT is a modified pretrained BERT network with Siamese and triplet network structures to produce meaningful sentence embeddings. The Siamese network architecture yields vectors of a fixed size for each input sentence and achieves state-of-the-art results for five out of seven tasks on SentEval (Conneau and Kiela, 2018), a popular toolkit to evaluate the quality of sentence embeddings. SBERT beats other contextual sentence embeddings like Universal Sentence Encoder (USE) and InferSent in performance and is computationally more efficient and faster.

Once the sentences are embedded, we can leverage traditional machine learning approaches for the classification task. We use the annotations for preferred app (CURRENT, OTHER, and NONE) as labels and train a traditional supervised classification models on comparative sentences from Pixie. We experiment with AdaBoost (Hastie et al., 2009), Random Forest (Breiman, 2001), and Support Vector Machines (SVM) (Chang and Lin, 2011).

3.3.2 Transformer Based Approaches

The Transformer (Vaswani et al., 2017) is a deep learning model architecture based on multi-head self attention. Transformers provide advantages over other deep learning architectures including standard recurrent neural networks (RNN), long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), and gated recurrent unit (GRU) networks (Chung et al., 2014). Recurrent models typically process sequences serially, which precludes parallelization while training and is a

Table 6: Masking app names in the sentence.

	Original sentence	**Masked sentence**
1	*CNN* should leave journalism to the pros at *Fox* news.	<*current_app*> should leave journalism to the pros at <*other_app*> news.
2	way better than *Pandora* by a long shot!!!!	way better than <*other_app*> by a long shot!!!!
3	*This* is a great game just like *Temple run*	<*current_app*> is a great game just like <*other_app*>

limitation for long sequences. Multi-head self attention in transformers provides greater parallelization and reduces training time while jointly attending to information at different positions in long sequences.

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) is a transformer-based deep learning language representation model. The BERT framework includes two steps, *pretraining* and *fine-tuning*, to address a wide variety of natural language processing (NLP) tasks. BERT can be pretrained on unsupervised tasks, 1. masked token prediction, and 2. next sentence prediction (NSP). The masked token prediction task is inspired by the cloze test (Taylor, 1953) and involves masking (replacing with a predefined tag) some percentage of tokens (words) in the text. BERT pretraining involves predicting a masked token given other tokens around it (context). This setup enables BERT to learn by optimizing a loss function based on the difference between the predicted and the actual tokens. The NSP task is related to representation-learning objective (Jernite et al., 2017; Logeswaran and Lee, 2018) and involves predicting if a sentence can be the next sentence given the current sentence.

BERT is pretrained on large text corpora including the BooksCorpus (800 million words) (Zhu et al., 2015) and English Wikipedia (2,500 million words). The pretrained BERT model can be fine-tuned for specific NLP tasks, such as question answering, semantic similarity, paraphrase detection, text entailment, and other natural language inference tasks. Fine-tuning provides considerable performance improvements even with a small labeled dataset, eliminating the need for large labeled datasets for each task. BERT advances the state-of-the-art for eleven benchmark NLP tasks, including the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018a), Stanford Question Answering Dataset (SQuAD v1.1 and v2.0) (Rajpurkar et al., 2016, 2018) and Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2018).

A replication study of BERT pretraining that measures the impact of multiple hyperparameters and training data sizes (Liu et al., 2019) find that the BERT model is highly under-trained. Robustly Optimized BERT Pretraining Approach (RoBERTa) (Liu et al., 2019) achieves further improvements on BERT on both GLUE and SQuAD benchmarks. Compared to BERT, RoBERTa is trained on more data, with longer sequences, bigger batches, for a longer time and does not include the next sentence prediction objective during pretraining.

Decoding-enhanced BERT with disentangled attention (DeBERTa) (He et al., 2020) further improves BERT and RoBERTa models with a disentangled attention mechanism and an enhanced mask encoder. DeBERTa represents each word with two vectors that encode its content and position, and leverages a new virtual adversarial training method for fine-tuning to improve the model's generalization. DeBERTa outperforms BERT and RoBERTa on several NLP tasks including MNLI, SQuAD, and RACE (Large-scale Reading Comprehension Dataset From Examinations) (Lai et al., 2017).

Increasing the model size in pretrained language representations often results in improved performance on downstream tasks, although at a higher computation cost, GPU/TPU memory, and longer training times. ALBERT (A Lite BERT) (Lan et al., 2019) takes advantage of parameter reduction techniques to lower memory consumption and increase the training speed of BERT. ALBERT scales better than the original BERT model and achieves comparable results on the GLUE, RACE, and SQuAD benchmarks. The ALBERT configuration similar to BERT-large has fewer parameters and trains much faster.

DistilBERT (Sanh et al., 2019) is a smaller, faster, and lighter version of BERT model that employs knowledge distillation during the pretraining phase to reduce the model size. DistilBERT is

smaller than BERT by 40%, while still retaining 97% of its language understanding capabilities, and is 60% faster.

BERT adopts an auto-encoding based pretraining that does not perform explicit density estimation. Masked tokens during BERT's pretraining are absent from real data at the time of fine-tuning, resulting in a pretrain-finetune discrepancy. Moreover, since the predicted tokens are masked in the input during pretraining, BERT assumes the predicted tokens are independent of each other given the unmasked tokens, which is an oversimplification. To overcome these limitations, XLNET (Yang et al., 2019), a generalized auto-regressive method, combines the auto-encoding and auto-regression approaches. XLNet outperforms BERT on 20 NLP benchmark tasks, including GLUE, SQuAD, and RACE.

We experiment with these transformer-based pretrained language models. Pretrained language models can be trained for specific tasks (such as classification) by adding a linear layer on top of the final hidden output layer. We can then apply a softmax function to the output of this linear layer to compute a probability distribution for the sentence over all classes. We fine-tune each pretrained model with Pixie to identify the preferred entity in a comparative sentence from app reviews. We adopt the Adam Optimizer with a 5e-5 learning rate, a weight decay of 0.01, and fine-tune each model for 20 epochs. We implement these models and fine-tune them with the Huggingface[2] library.

3.3.3 Entity-Aware Dependency-Based Deep Graph Attention Network (ED-GAT)

Ma et al. (2020) propose a dependency based attention model that achieves the state-of-the-art for the task of comparative preference classification. ED-GAT uses a multi-hop graph attention network over dependency graph to identify preferred entity in a comparative sentence. They use the Graph Attention Networks (GATs) (Veličković et al., 2017) to fuse the graph-structured information from the dependency graph and node features within the model. GAT has self attention layers allowing a node to attend to neighborhood features and learn different attention weights for different neighboring nodes. ED-GAT achieved a micro F1-score of 87.43% in identifying the preferred entity in a comparative sentence.

[2] https://huggingface.co/

To compare our results against ED-GAT, we convert the sentences in Pixie dataset into explicit comparisons. We follow the format of the CompSent-19 dataset (Panchenko et al., 2019). For explicit comparisons, if the first entity is the current app, we convert the label CURRENT to BETTER and OTHER to WORSE. If the first entity is the other app, we convert the label OTHER to BETTER and CURRENT to WORSE. For implicit comparisons, we add a token ([THIS]) for the current app in the front of the sentence, and convert the label CURRENT to BETTER and OTHER to WORSE. NONE labels stay the same.

We consider ED-GAT(Ma et al., 2020) as a baseline for the CPC task. ED-GAT requires dependency-parsing of the sentence to create a graph of the words. For implicit comparisons, we first parse the original sentence, and then create an edge between the added [THIS] token to the root of the sentence. We implement ED-GAT with BERT embeddings and 8 GAT layers.

4 Results

Table 7 contains results for models trained and tested on Pixie. Among the traditional approaches we experimented with, SVM yields the highest weighted F1-score of 71.86%, and precision and recall of 72.19% and 73.0%, respectively. Among transformer-based models, RoBERTa-large achieves highest weighted F1-score (85.23%), and RoBERTa-base and DeBERTa-base slightly trailing behind with F1-score of 84.26% and 83.73%, respectively. Overall, RoBERTa-large yields highest precision and recall scores for each individual class (except recall for the OTHER class for which DeBERTa achieves slightly higher recall of 88.51%).

All models we experimented with are more accurate with sentences where the current app is preferred than the ones where no app is preferred (or where the preference is unclear). This observation is reflected in the low F1-scores for both traditional and transformer-based approaches for the NONE class. For the NONE class, we achieve a highest F1-score of 46.39% for a traditional models (SVM) and 64.92% for a transformer-based model (RoBERTa-large). Recall for the NONE class is consistently lower than precision for each model we experimented with.

The highest F1-score for the CURRENT class with traditional models is 79.49% for SVM and 90.75% for transformer-based models with

Table 7: Results for preference classification on Pixie. The values are in % and the bold values highlight the highest numbers for each approach.

Approach	Model	CURRENT			NONE			OTHER			Weighted Average		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Traditional Approach	AdaBoost	71.57	73.44	72.49	45.06	35.29	39.58	63.53	68.30	71.57	63.80	64.62	64.07
	Random Forest	71.49	81.30	76.08	**64.80**	25.08	36.16	66.13	75.04	70.30	68.31	68.79	66.71
	SVM	**76.99**	**82.17**	**79.49**	62.63	**36.84**	**46.39**	**71.04**	**79.63**	**75.09**	**72.19**	**73.00**	**71.86**
Transformer Based Approach	DistilBert$_{base}$	82.49	87.53	84.94	61.23	52.32	56.43	80.65	80.40	80.52	77.95	78.52	78.14
	ALBERT$_{base\text{-}v2}$	87.30	87.41	87.35	58.86	57.59	58.22	82.70	83.46	83.08	80.44	80.54	80.49
	XLNET$_{base\text{-}cased}$	85.80	91.15	88.39	66.03	53.56	59.15	83.73	85.15	84.43	81.45	82.11	81.63
	RoBERTa$_{base}$	87.81	92.52	90.10	68.13	57.59	62.42	87.42	88.36	87.89	84.09	84.65	84.26
	RoBERTa$_{large}$	**88.60**	**93.02**	**90.75**	**68.99**	**61.30**	**64.92**	**88.75**	88.21	**88.48**	**85.09**	**85.49**	**85.23**
	DeBERTa$_{base}$	87.97	91.15	89.53	67.64	57.59	62.21	86.01	**88.51**	87.25	83.56	84.08	83.73
Prior work	ED-GAT	**83.24**	**78.05**	**80.57**	**48.89**	**54.49**	**51.54**	**76.28**	**77.79**	**77.03**	**74.44**	**73.68**	**73.99**

RoBERTa-large. For this class, the traditional approaches consistently yield a lower precision than recall while transformer-based models consistently achieve higher recall than precision. For the OTHER class, SVM achieves the highest F1-score (75.09%) among the traditional approaches, and RoBERTa-large (F1-score 88.48%) among the transformer-based models.

The ED-GAT model (Ma et al., 2020) trained on Pixie achieves a weighted average F1-score of 73.99%. For the CURRENT class it achieves an F1-score of 80.57% and for the NONE class it achieves an F1-score of 51.54%. This behavior is consistent with other transformer based model's results, all of which perform more accurately on the CURRENT class than on the NONE class.

We further analyze how each model performs on implicit and explicit comparative sentences separately. We train with both implicit and explicit comparative sentences but test each comparison type separately. Table 8 shows the results. RoBERTa yields the highest precision, recall, and f1 scores for both types of comparison (weighted f1-score of 84.40% for implicit and 86.26% for explicit comparisons). ED-GAT achieves an f1-score of 73.95% and 73.87% for implicit and explicit comparisons, respectively.

Transformer-based models achieve equal or higher numbers for explicit comparisons compared to implicit ones, whereas traditional models show a mixed pattern. For example, Random Forest yields higher f1-score on implicit comparisons while SVM achieves higher numbers on explicit comparisons.

5 Discussion and Future Work

We compare our results with the existing state-of-the-art method in preference classification, ED-GAT (Ma et al., 2020). ED-GAT achieves a weighted average F1-score of 87.43% on the CompSent-19 dataset in identifying the preferred entity in a comparative sentence. However, ED-GAT trained on CompSent-19 (Panchenko et al., 2019) dataset delivers a poor performance (F1-score 49.71%) when tested on Pixie. The model's performance improves substantially (weighted average F1-score 73.99%) when it is trained and tested on Pixie. This improvement could be because the CompSent-19 dataset lacks implicit comparisons, which is the majority class in Pixie.

In all our experiments, the models struggled the most in identifying the NONE class. Their poor performance could be because the NONE class includes both sentences that do not express a preference and sentences that express mixed preferences where it is difficult to determine a clear preference. This type of sentences were also the most ambiguous for manual annotation.

User generated text, such as app reviews, is typically different from formal text like news articles and scientific documents in terms of writing styles, including sentence structures and vocabulary. Comparative sentences in Pixie are limited to user app reviews and may not generalize well to other types of text. More analysis is needed to determine the generalizability of our dataset and the performance of a model trained on Pixie in identifying preferred entity in text from other domains.

A comparative sentence from user reviews that reveals a preference between two entities often includes the aspect of the comparison. For example,

Table 8: Results for preference classification for different types of comparisons. The values are in % and the bold values highlight the highest numbers for each type of comparison.

Type of Comparison	Model	Prec	Rec	F1
Implicit Comparison	AdaBoost	64.08	65.14	64.49
	Random Forest	70.65	71.41	68.98
	SVM	69.34	69.64	68.72
	DistilBert$_{base}$	78.16	78.78	78.39
	ALBERT$_{base\text{-}v2}$	80.12	80.58	80.32
	XLNet$_{base}$	80.33	81.18	80.65
	RoBERTa$_{base}$	83.51	84.21	83.74
	RoBERTa$_{large}$	**84.22**	**84.76**	**84.40**
	DeBERTa$_{base}$	84.09	**84.76**	84.28
	ED-GAT	74.21	73.80	73.95
Explicit Comparison	AdaBoost	63.47	63.95	63.54
	Random Forest	69.74	68.73	66.91
	SVM	74.41	75.60	74.30
	DistilBert$_{base}$	77.65	78.17	77.79
	ALBERT$_{base\text{-}v2}$	80.62	80.49	80.54
	XLNet$_{base}$	82.93	83.33	82.90
	RoBERTa$_{base}$	84.76	85.14	84.88
	RoBERTa$_{large}$	**86.17**	**86.43**	**86.26**
	DeBERTa$_{base}$	83.01	83.20	83.02
	ED-GAT	74.47	73.51	73.87

consider the following review on Uber Eats, *Uber eats app is way better than its competitors like Postmates and DoorDash; it is a better format and user friendly*. In addition to stating the preferred entity (Uber Eats), the reviewer specifies the reason for this preference by mentioning the aspect of comparison. The user prefers Uber eats over its competitors (postmates and DoorDash) in terms of format and user friendliness. Such feature-based comparative reviews are useful for consumers to make purchasing decisions when choosing between similar products. A feature-based study can also benefit app developers by informing what functionalities users like or dislike and what should be added or improved. Identify aspects of comparison in a comparative sentence and extracting user expectations from such feature-based comparisons are potential directions for future work.

6 Conclusion

We present Pixie, a new dataset with identified preferences in implicit and explicit comparisons in app review sentences. Pixie is manually annotated for preference classification and consists of 8,990 implicit and explicit comparative sentences. Pixie includes comparative sentences that have been overlooked by earlier work on preference classification, such as comparative sentences that omit the mention of an entity under comparison (implicit comparisons) or lack comparative linguistic cues (indirect comparisons). Compared entities are masked with predefined tags in Pixie to enable training of a general purpose preference classifications model for user reviews or other similar text.

We have found that the current state-of-the-art method for preference classification trained on Pixie outperforms traditional machine learning models, but lags behind transformer-based pretrained models fine-tuned on the dataset in terms of precision, recall, and F1 scores.

References

Bruce W. Ballard. 1988. A general computational treatment of comparatives for natural language question answering. In *Proceedings of the 26th Annual Meeting on Association for Computational Linguistics*, ACL '88, page 41–48, USA. Association for Computational Linguistics.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Joan W. Bresnan. 1973. Syntax of the comparative clause construction in english. *Linguistic inquiry*, 4(3):275–343.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.

Judith A. Chevalier and Dina Mayzlin. 2006. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345–354.

Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv*, abs/1412.3555.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 17th Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Wenjing Duan, Bin Gu, and Andrew B Whinston. 2008. The dynamics of online word-of-mouth and product sales—an empirical investigation of the movie industry. *Journal of Retailing*, 84(2):233–242.

Ronen Feldman, Moshe Fresko, Jacob Goldenberg, Oded Netzer, and Lyle Ungar. 2007. Extracting product comparisons from discussion boards. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 8001:469–474.

Carol Friedman. 1989. A general computational treatment of the comparative. In *27th Annual Meeting of the Association for Computational Linguistics*, pages 161–168, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Murthy Ganapathibhotla and Bing Liu. 2008. Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 241–248, Manchester, UK. Coling 2008 Organizing Committee.

Samir Gupta, A.S.M. Ashique Mahmood, Karen Ross, Cathy Wu, and K. Vijay-Shanker. 2017. Identifying comparative structures in biomedical text. In *Proceedings of the 16th Workshop on Biomedical Natural Language Processing*, pages 206–215, Vancouver, Canada,. Association for Computational Linguistics.

Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. 2009. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint*, abs/2006.03654.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168–177, New York, NY, USA. Association for Computing Machinery.

Yacine Jernite, Samuel R Bowman, and David Sontag. 2017. Discourse-based objectives for fast unsupervised sentence representation learning. *arXiv preprint*, abs/1705.00557.

Nitin Jindal and Bing Liu. 2006. Identifying comparative sentences in text documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 244–251, New York, NY, USA. Association for Computing Machinery.

Wiltrud Kessler and Jonas Kuhn. 2013. Detection of product comparisons - how far does an out-of-the-box semantic role labeling system take you? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1892–1897, Seattle, Washington, USA. Association for Computational Linguistics.

Wiltrud Kessler and Jonas Kuhn. 2014. A corpus of comparisons in product reviews. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*, pages 2242–2248, Reykjavik, Iceland. European Language Resources Association (ELRA).

Takeshi Kurashima, Katsuji Bessho, Hiroyuki Toda, Toshio Uchiyama, and Ryoji Kataoka. 2008. Ranking entities using comparative relations. In *Database and Expert Systems Applications*, pages 124–133, Berlin, Heidelberg. Springer Berlin Heidelberg.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint*, abs/1909.11942.

Kevin Lerman and Ryan McDonald. 2009. Contrastive summarization: An experiment with consumer reviews. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 113–116, Boulder, Colorado. Association for Computational Linguistics.

Yuanchun Li, Baoxiong Jia, Yao Guo, and Xiangqun Chen. 2017. Mining user reviews for mobile app comparisons. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint*, abs/1907.11692.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint*, abs/1803.02893.

Nianzu Ma, Sahisnu Mazumder, Hao Wang, and Bing Liu. 2020. Entity-aware dependency-based deep graph attention network for comparative preference classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5782–5788, Online. Association for Computational Linguistics.

Nielsen. 2012. Consumer trust in online, social and mobile advertising grows. Technical report, Nielsen. [Online; accessed: 2021-10-28].

Nielsen. 2020. The data behind word of mouth marketing. Technical report, Wooly. [Online; accessed: 2021-10-28].

Alexander Panchenko, Alexander Bondarenko, Mirco Franzek, Matthias Hagen, and Chris Biemann. 2019. Categorizing comparative sentences. In *Proceedings of the 6th Workshop on Argument Mining*, pages 136–145, Florence, Italy. Association for Computational Linguistics.

Alexander Panchenko, Eugen Ruppert, Stefano Faralli, Simone P. Ponzetto, and Chris Biemann. 2018. Building a web-scale dependency-parsed corpus from CommonCrawl. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Dae Hoon Park and Catherine Blake. 2012. Identifying comparative claim sentences in full-text scientific articles. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pages 1–9, Jeju Island, Korea. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Manny Rayner and Amelie Banks. 1988. Parsing and interpreting comparatives. In *26th Annual Meeting of the Association for Computational Linguistics*, pages 49–60, Buffalo, New York, USA. Association for Computational Linguistics.

Manny Rayner and Amelie Banks. 1990. An implementable semantics for comparative constructions. *Computational Linguistics*, 16(2):86–112.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint*, abs/1910.01108.

Steffen Staab and Udo Hahn. 1997. Comparatives in context. In *Proceedings of the 14th National Conference on Artificial Intelligence*, AAAI'97/IAAI'97, page 616–621, Providence, Rhode Island. AAAI Press.

Arnim von Stechow. 1984. Comparing semantic theories of comparison. *Journal of Semantics*, 3(1-2):1–77.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Robert van Rooij. 2011. *Implicit versus Explicit Comparatives*, pages 51–72. Palgrave Macmillan UK, London.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, Long Beach, CA. Curran Associates, Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *Sixth International Conference on Learning Representations*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wenxin Wang, Yi Feng, and Wenqiang Dai. 2018b. Topic analysis of online reviews for two competitive products using latent dirichlet allocation. *Electronic Commerce Research and Applications*, 29:142–156.

Adina Williams, Nikita Nangia, and Samuel Bowman. 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)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Kaiquan Xu, Stephen Shaoyi Liao, Jiexun Li, and Yuxia Song. 2011. Mining comparative opinions from customer reviews for competitive intelligence. *Decision Support Systems*, 50(4):743–754. Enterprise Risk and Security Management: Data, Text and Web Mining.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, Vancouver. Curran Associates, Inc.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, Santiago, Chile. IEEE.