Distilling Salient Reviews with Zero Labels

Chieh-Yang Huang[1]*, Jinfeng Li[2], Nikita Bhutani[2],
Alexander Whedon[3]†, Estevam Hruschka[2], Yoshihiko Suhara[2]
Pennsylvania State University[1], Megagon Labs[2], Stitch Fix[3]
chiehyang@psu.edu[1], {jinfeng, nikita, estevam, yoshi}@megagon.ai[2], alexander.whedon@gmail.com[3]

Abstract

Many people read online reviews to learn about real-world entities of their interest. However, majority of reviews only describes general experiences and opinions of the customers, and may not reveal facts that are specific to the entity being reviewed. In this work, we focus on a novel task of mining from a review corpus sentences that are unique for each entity. We refer to this task as **Salient Fact Extraction**. Salient facts are extremely scarce due to their very nature. Consequently, collecting labeled examples for training supervised models is tedious and cost-prohibitive. To alleviate this scarcity problem, we develop an unsupervised method **ZL-Distiller**, which leverages contextual language representations of the reviews and their distributional patterns to identify salient sentences about entities. Our experiments on multiple domains (hotels, products, and restaurants) show that ZL-Distiller achieves state-of-the-art performance and further boosts the performance of other supervised/unsupervised algorithms for the task. Furthermore, we show that salient sentences mined by ZL-Distiller provide unique and detailed information about entities, which benefit downstream NLP applications including question answering and summarization.

1 Introduction

Online reviews have become a rich source of information for people to know more about real-world entities for making purchasing decisions (BrightLocal, 2019). Reviews contain diverse information ranging from general sentiments and customer experiences to features and attributes about an entity. Table 1 shows examples of different types of information found in reviews. Since consuming a large number of reviews can be cumbersome, text mining tools and algorithms are popularly used to uncover and aggregate customer sentiments expressed in opinions and experiences to provide a summary of how the entities are perceived by customers. However, existing mining tools largely ignore information about *unique* features and attributes of the reviewed entity. Such information tends to be sparse compared to expressions about usage, experience and opinions. We observe that in domains such as hotel reviews, sentences with unique features can be as few as 5% of all sentences in the reviews. In a public dataset (Reviews, 2021), for example, "rooftop bar" of Table 1 appears in only 3,026 of 8,211,545 sentences and the attribute is rare that exists in only 197 of 3945 TripAdvisor hotels. Nevertheless, such information is of great interest to users and can be further useful for many downstream applications such as ranking reviews, creating concise entity summaries and answering questions about the entities.

In this work, we focus on mining sentences that describe unique information about entities from its reviews. We call these unique sentences salient facts and denote this task as **Salient Fact Extraction**. Although scarce, salient facts exhibit at least one of the two characteristics: (a) they mention attributes rarely used to describe other entities (example 1 in Table 1), or (b) they convey unique, detailed information (e.g. numeric or categorical) about a common attribute (example 2 in Table 1). Due to the scarcity of salient facts in the reviews, collecting a labeled dataset to train a supervised model is extremely inefficient and cost-prohibitive.

Although there is a rich body of research on extracting tips, informative and helpful sentences from reviews (Li et al., 2020; Novgorodov et al., 2019; Negi and Buitelaar, 2015; Guy et al., 2017a; Wang et al., 2019; Chen et al., 2014; Hua et al., 2019; Zhang et al., 2019; Gao et al., 2018), these approaches have several limitations for extracting

*Work done during internship at Megagon Labs.

†Work done while at Megagon Labs.

The Fifth Workshop on Fact Extraction and VERification (FEVER). Co-located with Association for Computational Linguistics 2022.

	Sentence	Type
1	There is a rooftop bar.	Salient Fact
2	The hotel gives 90% discount for seniors.	Salient Fact
3	The price is cheap.	Sentiment
4	We stayed 3 nights here.	Usage Experience
5	Choose other hotels instead.	Suggestion

Table 1: Different types of information in hotel reviews. A salient fact mentions attributes (marked in blue) distinctive to the hotel or provides uncommon descriptions (marked in red) for common attributes.

salient facts. Firstly, informativeness and saliency are related but have subtle differences. Not all informative sentences describe unique information about an entity. Secondly, due to scarcity of salient facts, collecting labeled training data to train supervised techniques (which is the common technique used for finding informative reviews) can be expensive and time-consuming.

To address the scarcity problem, we propose a novel unsupervised extractor for identifying salient sentences in a zero-label setting where abundant unlabeled reviews are available. A naive approach is to refer to the distributional patterns of salient sentences in a review corpus. We projected all the sentences in a corpus to a t-SNE plot (Hinton and Roweis, 2002) and found that salient sentences tend to appear as border points on the graph. However, we observed that not all border points are salient facts. Many sentences mentioning named entities names or unique personal stories also appear as border points. Such non-informative sentences thus make distributional patterns noisy and the extraction challenging.

Based on these distributional patterns, we propose a novel system, ZL (Zero Label) - Distiller, which uses two Transformer-based models for capturing unique and informative distributional patterns to extract salient facts. It uses a Transformer-based entity prediction model to identify most `unique` sentences for an entity, and another Transformer-based model to filter out non-informative sentences, such that `informative` sentences can be kept. The former one measures how distinctive a review sentence is to the corresponding entity but not to others. The latter one masks entity names in all sentences and drops those sentences that are likely personal stories. To our best knowledge, this is the first work to capture distributional patterns of all sentences for mining useful review sentences.

Contributions. In summary, our contributions are four folds. (1) We formulate a novel task that extracts entity-specific information (denoted as salient facts) from online reviews (2) To deal with scarcity of salient facts, we present an unsupervised method ZL-Distiller, which relies on distributional patterns instead of human annotations. (3) We show that ZL-Distiller leads to new state-of-the-art performance when used independently, or combined with supervised models on 3 domains (Hotel, Product and Restaurant). (4) We demonstrate that ZL-Distiller benefits downstream applications including question answering, and entity summarization by removing non-informative sentences from the pipeline.

2 Related Work

Helpful review definitions. Research community has continuously devoted to understanding which reviews are the most helpful (Li et al., 2020). The gold standard is to collect labels (e.g. helpful or not helpful votes) from various readers passively. Recently, researchers begin to realize that helpful reviews are broad, so they proactively propose sub-concepts, including tip(Hirsch et al., 2021; Guy et al., 2017a; Challenge, 2020), suggestion (Negi and Buitelaar, 2015; Negi et al., 2019; Moghaddam, 2015), and sentiment (Liu, 2012), as complements. To further address this issue, we introduce salient facts as a novel sub-concept, that aims at extracting the most entity-specific information from raw reviews. We demonstrate the real value of salient facts through three natural language processing applications, including saliency estimation, question answering, and entity summarization. Similar to existing sub-concepts, we anticipate the widespread adoption of salient facts in various domains, including but not limited to hotel (Negi and Buitelaar, 2015), product (Novgorodov et al., 2019), restaurant (Challenge, 2020), and travel (Guy et al., 2017a), in the near future.

Label-reliant solutions. Most of existing extraction models (Novgorodov et al., 2019; Negi and Buitelaar, 2015; Guy et al., 2017a; Wang et al., 2019; Chen et al., 2014; Hua et al., 2019; Zhang et al., 2019; Gao et al., 2018; Li et al., 2019; Evensen et al., 2019) are supervised. Although their extraction qualities approximate human performance, the deployment of these models requires a great amount of human labels. Collecting labels for these models can be time-consuming and costly since the process deals with worker education, salary negotiation, and mistake label filtering. Therefore, we propose ZL-Distiller that adopts a label-free design choice while is also compatible to label-reliant solutions.

Label-free solutions. Some label-free solutions attempted to remove the reliance on labels by leveraging data characteristics. For example, Zero-shot learning (Lewis et al., 2020) predicts a sentence as true if its embedding is close to the class name (e.g. salient fact or helpful). Unsupervised entity extraction (Akbik et al., 2018, 2019a,b; Schweter and Akbik, 2020) predicts the sentence as true if its tokens contain named entities, such as person or location. Though these methods have sufficiently leveraged lexical characteristics of a single sentence, they are incapable of leveraging common characteristics of a group of sentences (e.g. salient facts), with which helpful review mining can be substantially boosted. Our label-free solution, i.e. ZL-Distiller, identifies two distributional patterns of salient facts, i.e. `unique` and `informative`, to extract the comments containing salient facts. By utilizing these characteristics, ZL-Distiller shows superior performance in the salient fact extraction task.

3 Method

A summary of ZL-Distiller and its performance comparison with existing systems is depicted in Figure 1. Overall, ZL-Distiller is an unsupervised extractor that leverages distributional patterns (Figure 1A) to identify salient facts. ZL-Distiller introduces two components, `Unique` model and `Informative` model (Figure 1B and upper panel of 1C), to predict the uniqueness of a sentence and to exclude non-informative sentences, respectively. ZL-Distiller achieves better performance when compared with unsupervised baselines (e.g. zero-shot learning) under unsupervised setting (Figure 1B). Though ZL-Distiller shows worse performance compared with supervised baselines (upper panel of Figure 1C), it boosts the performance when used jointly with supervised solutions (e.g. BERT) under supervised setting (lower panel of Figure 1C).

3.1 Salient Fact Extraction

We formulate **Salient Fact Extraction** as a sentence classification task. We choose a sentence to be an instance instead of a review because a review could contain both relevant and irrelevant content. Giving a set of entities $\mathbf{E} = \{e_1, e_2, \cdots, e_i, \cdots, e_n\}$ with n different entities in a specific domain, each entity would have its own set of review sentences $\mathbf{S}_i = \{s_{i,1}, s_{i,2}, \cdots, s_{i,j}, \cdots, s_{i,m}\}$, where $s_{i,j}$ means a review j sentence from e_i. Within $\mathbf{S}_i$, our goal is to find out review sentences that are representative for e_i compared with all other entities. As a sentence classification task, each review sentence $s_{i,j}$ will be given a label of $\{0, 1\}$, where 1 means salient fact. The set of n entities can be defined by their real-world affinity (e.g. hotels on the same street or companies of the same field).

3.2 `Unique` Model

We notice that a salient fact review sentence means that the sentence should be (*i*) representative for the corresponding entity, (*ii*) unique for the corresponding entity, and (*iii*) not applicable for other entities. Figure 2 shows the idea. For Entity 1 in Figure 2, we can separate all the review sentences into two groups, (A) sentences that are representative and unique for Entity 1 and (D) sentences that are also applicable to Entity 2 and 3. Given the idea, our goal is to extract review sentences in (A). And such extraction strategy can be applied to any number of entities. In order to find out salient review sentences, we will need to model the distribution of the review sentences for each entity. By comparing the distribution, we can design a scoring function to rank the level of saliency.

3.2.1 Distribution Modeling

We fine-tune BERT (Devlin et al., 2019) to model the distribution of the review sentences. The model is designed as a multi-class classifier where each class stands for an entity e_i. We first feed the whole review sentence into BERT. On top of the representation of `[CLS]`, we apply a dense layer and a softmax function to get the probability over the entities. The probability $P(e_i|s_{i,j})$ outputted by the model is then the estimated probability of a $sentences_{i,j}$ belonging to entity e_i. Notice that higher probability also means that the review sentence is more representative for entity e_i.

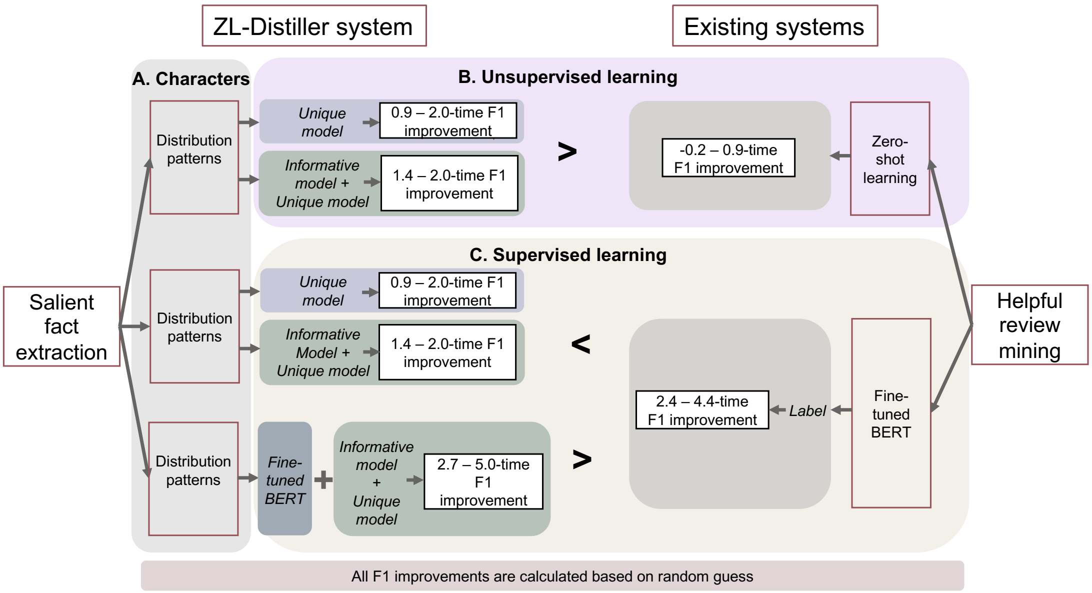

ZL-Distiller system
Existing systems
A. Characters
B. Unsupervised learning
Distribution patterns
Unique model
0.9 – 2.0-time F1 improvement
Informative model + Unique model
1.4 – 2.0-time F1 improvement
>
-0.2 – 0.9-time F1 improvement
Zero-shot learning
C. Supervised learning
Salient fact extraction
Distribution patterns
Unique model
0.9 – 2.0-time F1 improvement
Informative Model + Unique model
1.4 – 2.0-time F1 improvement
<
Helpful review mining
2.4 – 4.4-time F1 improvement
Label
Fine-tuned BERT
Distribution patterns
Fine-tuned BERT
+
Informative model + Unique model
2.7 – 5.0-time F1 improvement
>
All F1 improvements are calculated based on random guess

Figure 1: Summary of this paper. Fine-tuned BERT with ZL-Distiller achieves the best F1 improvement. Here when we compare all systems against the same baseline, random guess, that predicts a review as salient at the probability of %positive (the ratio of salient facts shown in Table 3).

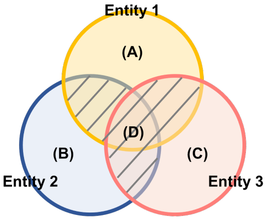


Figure 2: The idea of ZL-Distiller. The reviews of every entity can be separated into two parts, unique and representative sentences that are only applicable for a specific entity and sentences describing facts that share with other entities. ZL-Distiller will extract review sentences that are unique to every entity. For example, review comments given to Entity 1: "(A) The hotel provides free shuttle from/to the airport. (D) I like this hotel". Review comments given to Entity 2: "(B) the hotel is the tallest building with awesome views. (D) I really like this one". Review comments given to Entity 3: "(C) This is the only hotel that offers free parking. (D) A perfect place to live in." For Entity 1, ZL-Distiller will automatically extract the sentence (A) containing salient facts from comments.

3.2.2 Scoring Design

Given the estimation of probability, we design the following scoring function to find out review sentences that are representative for entity e_i but not applicable for other entities $\mathbf{E} - e_i$:

$$Score(s_{i,j}) = P(e_i|s_{i,j}) - \frac{1}{|\mathbf{E} - e_i|} \sum_{e_k \in \mathbf{E} - e_i} P(e_k|s_{i,j}) \tag{1}$$

The higher value of the first term $P(e_i|s_{i,j})$ measure if $s_{i,j}$ is representative for its own entity e_i. The second term $\frac{1}{|\mathbf{E}-e_i|} \sum_{e_k \in \mathbf{E}-e_i} P(e_k|s_{i,j})$ measures whether the $s_{i,j}$ is also applicable to other entities. Overall, the range of the score is between -1 to 1 with 1 stands for the perfect case of salient facts.

3.3 `Informative` Model

We next design `Informative` Model to further improve extraction performance. We explore a set of techniques that can be summarized as two heuristics i.e. irrelevance removal and target name removal. The informative model output is fed as the input of the unique model.

3.3.1 Irrelevance Removal

As shown in Table 2, column "**Review Sentence**", some people would describe their own experience which is not necessarily relevant to the entity when writing reviews. Such irrelevant review sentences could be noises when training the model to estimate the entity distribution. Therefore, we train irrelevance classifiers as a binary classifier using BERT (Devlin et al., 2019) that can be used in different domains. The BERT was trained with 600 manually annotated sentences. These sentences were sampled from the same source as the salient facts datasets (Reference in Section 4.1). The review sentences that convey relevant information are labeled as 0, whereas those conveying irrelevant in-

Review Sentence	Label
Many are still buying the KXTG76xx.	0
I purchased this nice phone for my husband	1
The smaller handsets are the same size as the ATT sets I'm replacing from 8 years ago.	0
They have the same amount of volume too.	0
Could be louder but same volume as my iPhone.	0

Table 2: Example review sentences of relevance and irrelevance. Some people would describe something that is not necessarily relevant to the target entity. These irrelevant review sentences will be labeled as 1. Otherwise, relevant review sentences will be labeled as 0. Sentences are extracted from Amazon office product review dataset.

formation are labeled as 1. Given that we only have a few annotations, we split data into ten folds and train ten models where each model is trained on the selection of nine folds. Notice that even though the goal is to remove the irrelevant review sentences, accidentally removing a relevant review sentence is undesired. Therefore, we take a strict way to aggregate the models' output by averaging all the predicted probabilities. When applying irrelevance removal rule, review sentences that are predicted as *irrelevant* will be removed for both training and testing.

3.3.2 Target Name Removal

When writing reviews, it is highly possible to mention the name of the target entity, such as "*I stayed at the **Library Hotel** over Christmas and it was a true delight.*" and "*There are so many things about **The Library** that make it my new favorite hotel in NYC.*" It is obvious that when mentioning the target name of the entity, such review sentences will have high score as they are totally unique to the target entity and not applicable to other entities at all. We thus believe that target name removal is necessary. To do so, we turn the target name into a dummy symbol `[TARGET_NAME]`. However, as we can see in the above mentioned examples, people could refer to the target entity using different aliases such as "Library Hotel" or "The Library". Automatically extracting alias itself is a hard problem in natural language processing field.

To solve this problem, we gather all potential aliases of the targeted entity to augment the list of entity names before training. Notice that in some domains, it is infeasible to gather aliases as the target entity name is too general such as Product domain from Amazon review. During training stage, we feed the augmented list to ZL-Distiller so that it can maximally recognize the entity names. We cannot rule out the possibility that some rare entity aliases will be retained in the comments after target name removal, but most of aliases of the target entity will be removed. In our experiments, target name removal can bring up to 4.3% F1 performance improvement.

Domain	#Sample	#Positive	#Negative	%Positive
Hotel	1008	164	844	16.3%
Product	1015	69	946	6.8%
Restaurant	766	45	721	5.9%

Table 3: Dataset statistics.

4 Experiment

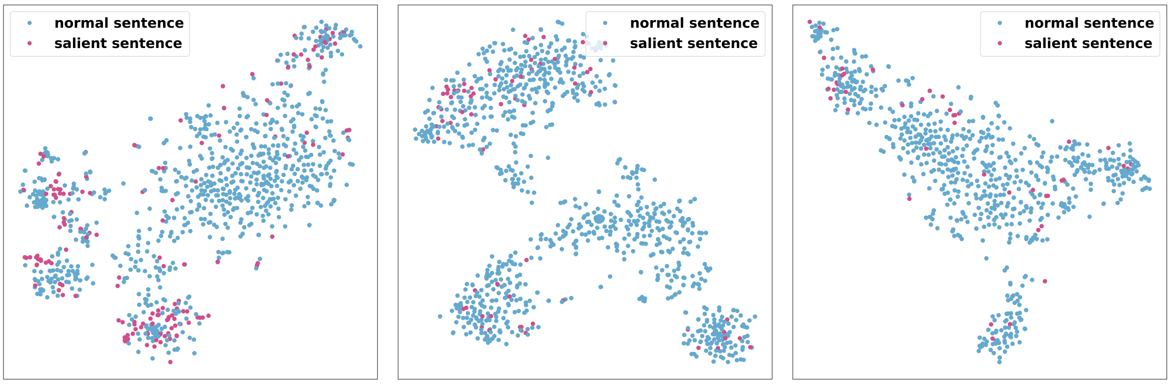
4.1 Datasets

We obtain Hotel, Product, and Restaurant datasets from public reviews of TripAdvisor (Reviews, 2021) [1], Amazon (He and McAuley, 2016), and Yelp [2], respectively. Since a review contains multiple sentences, we split every review into individual sentences using NLTK tokenizer.

We randomly sample 1008, 1015, and 766 sentences for Hotel, Product, and Restaurant, respectively. We invite human editors to label sentences, with label 1 representing the sentence containing a salient fact and label 0 otherwise. The cohen's kappa of two annotators is 0.80. The value indicates a high degree of agreement when compared with the results of existing helpful review annotation (e.g., 0.81 from suggestion annotation (Negi and Buitelaar, 2015) and 0.59 from travel tip annotation (Guy et al., 2017b)). The datasets statistics regarding three domains are shown in Table 3, and the full data annotation process is in Appendix, section Data Annotation.

Evaluation metric. We use F1 score, i.e. the harmonic mean of precision and recall [3], to evaluate the extraction performance. Since salient facts are sparse and dominant label is label 0, we use F1 scores of label 1 for accurate assessment (Li et al., 2020).

[1] https://www.cs.cmu.edu/~jiweil/html/hotel-review.html
[2] https://www.yelp.com/dataset/documentation/main
[3] https://en.wikipedia.org/wiki/F-score



(a) Hotel

(b) Product

(c) Restaurant

Figure 3: t-SNE plot of BERT [CLS] sentence embeddings (see Section 3.2.1). Semantically similar sentences appear close in the graph. Salient fact sentences tend to appear at borders, indicating they are dissimilar to normal sentences that appear at the center. Some normal sentences would also appear at borders, indicating unique sentences are not necessarily salient facts. The results suggest that we need both `Unique` and `Informative` models for the best extraction quality.

4.2 Distributional Patterns

To obtain distributional patterns of salient facts among all sentences, we use a t-SNE plot to visualize semantic similarity between different sentences. After being projected to the two-dimensional t-SNE plot, more similar sentences will appear closer in the graph. In our t-SNE analysis, we first input every sentence to BERT and use [CLS] vector as its vector representation. We next visualize all the vectors to the t-SNE plot, with salient facts marked in red and normal sentences marked in blue. The t-SNE plots for Hotel, Product, and Restaurant show clear patterns of salient facts distribution as shown in Figure 3.

On all the three t-SNE plots, salient facts tend to appear at borders but not centers. The pattern suggests that salient facts tend to provide unique information that is specific to the corresponding entity. This "unique" pattern motivates the design of `unique` model of ZL-Distiller. Besides, though border points are the most unique sentences, we notice that a large number of them are not salient facts. They appear at border not because they are salient, but because they contain uncommon words such as entity name or personal stories. Such uncommon words do not convey "informative" messages about the entity. Therefore, in addition to the `unique` model, ZL-Distiller adopts an `informative` model to mask entity names and drops personal stories sentences. Further analysis on the differences of salient facts and normal sentences regarding key phrases are in Appendix, section Explanation of Saliency with Key Phrases.

	Hotel	Product	Restaurant
Random guess	0.163	0.068	0.059
TextRank	0.309	0.146	0.100
LexRank	0.304	0.150	0.096
Zero-Shot	0.133	0.129	0.071
PacSum (bert)	0.273	0.127	0.070
PacSum (finetune)	0.240	0.200	0.079
PacSum (tfidf)	0.342	**0.317**	0.077
ZL-Distiller	**0.407**	0.201	**0.144**
ZL-Distiller + PacSum	**0.424**	**0.414**	**0.300**

Table 4: F1 score comparison with the state-of-the-art unsupervised baselines. Best scores are marked in bold. ZL-Distiller outperforms all baselines except PacSum (tfidf) on Product. ZL-Distiller further boosts the performance when combined with PacSum (tfidf). More performance results of ZL-Distiller + PacSum (tfidf) are in the Appendix, Section "Performance of Jointly Unsupervised Prediction".

The effects of `unique` and `informative` models on the extraction performance are in Appendix, section Effect of `Unique` Model and section Effect of `Informative` Model.

4.3 Comparison with Label-free Solutions

Zero-shot learning (HuggingFace, 2020; Lewis et al., 2020) is one of the state-of-the-art solutions that require zero training labels for text extraction. Zero-shot learning can predict the probability of the review belonging to the class, if it is fed with a review and a class name. Therefore, we apply zero-shot learning to the salient fact extraction task. Specifically, we iterate the class name in a set of "salient", "interesting", "informative", "unique",

Model	Top-n			
	10	30	50	100
	Hotel			
ZL-Distiller	0.222	0.338	0.400	0.415
BERT	**0.356**	0.585	**0.588**	**0.459**
ZL-Distiller +BERT	**0.356**	**0.615**	0.565	**0.459**
	Product			
ZL-Distiller	0.276	0.245	0.348	0.269
BERT	**0.345**	0.367	0.319	0.269
ZL-Distiller +BERT	**0.345**	**0.408**	**0.377**	0.**286**
	Restaurant			
ZL-Distiller	0.200	**0.400**	**0.267**	**0.182**
BERT	0.200	0.200	0.167	0.091
ZL-Distiller +BERT	0.200	0.300	**0.267**	**0.182**

Table 5: Performance of supervised learning (i.e. BERT and ZL-Distiller + BERT) using domain-specific labels. Best F1 scores are marked in bold.

and "concrete", and pick the class name that offers the best extraction performance. When evaluating a class name, we vary the prediction threshold from 1 to 0, and report the highest F1 score. We use HuggingFace implementation (HuggingFace, 2020) of zero-shot learning (Lewis et al., 2020) for experiments.

In addition to zero-shot learning, we also deploy popular text summarization algorithms, which are TextRank (Mihalcea and Tarau, 2004), LexRank (Erkan and Radev, 2004), and three variants of PacSum (bert/finetune/tfidf) (Zheng and Lapata, 2019) for comparison. These algorithms select informative sentences to represent long text so their outputs naturally form a set of candidate salient facts.

We present F1 scores of all methods in Table 4. According to the results, ZL-Distiller shows comparable or better performance compared with existing methods, consistently on Hotel, Product, and Restaurant. Furthermore, we combine the prediction scores of ZL-Distiller and PacSum (tfidf) by taking dot product for every sentence. We observe that such combination leads to the highest F1 scores on all three datasets. The results suggests that ZL-Distiller serves as a new strong baseline for salient fact extraction. Meanwhile, ZL-Distiller can work with existing baselines to achieve the best performance.

4.4 Performance of Jointly Supervised prediction

ZL-Distiller can extract salient facts in unsupervised manner where data labels are absent. Since ZL-Distiller captures distributional patterns, we then investigate whether ZL-Distiller is still useful in supervised manner where data labels are present. Briefly, we use BERT with data labels as the representative supervised solution. Next, we combine BERT prediction scores with ZL-Distiller prediction scores, by taking products of BERT and ZL-Distiller scores and, then, rank sentences by product scores. We denote this combination as ZL-Distiller + BERT. Finally, we take the top-*n* as the predicted salient facts and then return F1 scores when setting top-*n* with various number, i.e. 10, 30, 50, and 100.

We present the F1 scores of ZL-Distiller, BERT, and ZL-Distiller + BERT in Table 5. As expected, ZL-Distiller F1 scores are lower than BERT on Hotel and Product as ZL-Distiller does not use domain-specific labels. However, ZL-Distiller shows better performance than BERT on Restaurant. The reason is that Restaurant has extremely low ratio of salient facts (i.e. 5.9%, as shown in Table 3), for which the number of salient facts for training is insufficient. The results suggest that ZL-Distiller is effective when there are no or insufficient data labels.

When there are sufficient labels (e.g. on Hotel and Product), ZL-Distiller performs worse than supervised solution (i.e. BERT). However, ZL-Distiller is still helpful, indicated by the results that ZL-Distiller + BERT achieves better F1 scores than BERT on all three datasets. The highest F1 improvement is 10%, 18%, and 100%, on Hotel, Product, and Restaurant, respectively, as shown in Table 5. Such improvement is general to various domains and this is because that ZL-Distiller can always capture distributional patterns as discussed in Section 4.2.

5 Application

In this section, we demonstrate the effect of salient facts in downstream NLP applications. We apply salient fact extraction in company reviews, and select three downstream applications, including review saliency estimation, question answering, and company summarization. We used ZL-Distiller + BERT (denoted as saliency prediction model) to obtain salient facts as inputs for downstream applications.

5.1 Saliency Estimation

An important application of salient fact extraction is saliency estimation, which returns the probabilities of a text being salient and non-salient. To perform saliency estimation, we deploy our saliency prediction model to evaluate two reviews of Google

Google review	Pos.	Neg.
When a Google employee passes away, surviving spouse receives 50% of their salary for the next 10 years.	0.65	0.39
awesome place to work, great salary, smart people.	0.02	0.99

Table 6: Saliency estimation of a raw review in terms of saliency (i.e. Pos.) and non-saliency (i.e. Neg.) scores.

Question	Salient	Raw
How long is parental leave?	12 weeks	nice amount of leave
How much would company pay for health insurance?	90%	401k

Table 7: Question answering based on Google reviews using DistilBERT (Sanh et al., 2019).

and show the probabilities in Table 6. The first review reveals a rare company policy (i.e. death benefit) and numeric descriptions (i.e. 50% salary and 10 years), which are considered unique information. The model gives a higher probability of salient (i.e. 0.65) than non-salient (i.e. 0.39), suggesting that saliency prediction model can appropriately rank unique sentences. The second review discusses common attributes such as work, salary, and people and uses sentimental descriptions like awesome and great, which are considered as non-unique information. The model predicts a lower probability of salient (i.e. 0.02) than non-saliency (i.e. 0.99), suggesting that saliency prediction model can rank non-unique sentences. Taken together, these saliency estimation probabilities serve as good references for readers to select or rank raw reviews.

5.2 Question Answering

Question answering (QA) tasks (such as SQuAD 1.0 and 2.0), take a knowledge-seeking question and a text context as inputs and then retrieves answer for the question in the context. Though the process is straightforward, application of QA to reviews meets a challenge, which is widespread general comments (e.g. sentiments) that lead to wrong answers. To overcome this challenge, we use saliency prediction model to prioritize informative reviews. In brief, we prepare two contexts using different sentences (i.e. salient facts and raw reviews) and input two questions (i.e. "How long is parental leave" and "How much would company pay for health insurance") for both contexts.

Googlers can relax after a long day by braving the rock climbing wall, playing billiards, or just relaxing in a self-controlled massage chair. Google is paying out my unvested options and RSUs and gave me a grant of GSUs to boot.
Awesome place to work, great salary, smart people, lots of happy hours and the free food is as great as everyone says it is. Too much emphasis on work life balance. Can really make a difference in the world.

Table 8: Summary of Google using salient facts (up) and raw reviews (down). Salient facts enable finer-grained summarization that presents specific attributes (e.g. rock climbing wall) of Google rather than general attributes (e.g. work life balance) of Company class.

We then use HuggingFace question answering engine (Face, 2020; Sanh et al., 2019) to look for answers in contexts to obtain company knowledge. Our results show that salient facts context returns higher-quality answers than raw reviews context. For example, for the question "How long is parental leave", salient facts return an objective and unbiased answer (i.e. 12 weeks), whereas raw reviews return a subjective and biased answer (i.e. nice amount of leave). More comparative results are shown in Table 7. These results suggest that salient facts enable accurate question answering over reviews, where objective and subjective texts are mixed.

5.3 Entity Summarization

According to our results, salient facts represent a collection of unique sentences in the reviews. In addition to their uniqueness, we also find that salient facts can serve as ingredients for high-quality entity summarization. We compare two summaries of Google reviews (shown in Table 8) based on salient facts and raw reviews, respectively. We use BART (Lewis et al., 2020) as summarizer and set the expected number of words to 50. The results show that salient facts based summary is more specific to the entity as it reveals finer-grained attributes (e.g. rock climbing wall). Moreover, salient facts based summary is unbiased as it seldom contains sentimental words (e.g. awesome and great). Contrastively, raw reviews based summary mentions commonsense attributes (e.g. work and salary) and sentimental words (e.g. awesome and great) more frequently. Therefore, these results suggest that salient facts based summary will be more favorable for readers who are looking for informative and unbiased entity summarization.

6 Conclusion

In this paper, we propose to extract salient facts from online reviews. To achieve this goal, we develop ZL-Distiller, which is the first-of-its-kind system for salient fact extraction. ZL-Distiller does not require human labels, but labels can further boosts its performance. To prove that salient facts can be applied to popular real-world applications, we conduct a study on raw company reviews, which demonstrates that salient facts can improve the quality of downstream applications, including saliency estimation, question answering and company summarization. These results implicate the feasibility of salient fact extraction in real-world text corpus including company reviews, which consist of both salient and non-salient contents. Our practice suggests that the general-purpose salient fact extraction has a substantial effect on existing text-based applications for diverse domains.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019a. Flair: An easy-to-use framework for state-of-the-art nlp. In NAACL, pages 54–59.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019b. Pooled contextualized embeddings for named entity recognition. In NAACL, page 724–728.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In COLING, pages 1638–1649.

BrightLocal. 2019. https://www.brightlocal.com/research/local-consumer-review-survey/. Local Consumer Review Survey.

Yelp Dataset Challenge. 2020. https://www.yelp.com/dataset/documentation/main. In YELP.

Ning Chen, Jialiu Lin, Steven C. H. Hoi, Xiaokui Xiao, and Boshen Zhang. 2014. Ar-miner: mining informative reviews for developers from mobile app marketplace. In ICSE, pages 767–778.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT, pages 4171–4186.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. JAIR, 22:457–479.

Sara Evensen, Aaron Feng, Alon Y. Halevy, Jinfeng Li, Vivian Li, Yuliang Li, Huining Liu, George A. Mihaila, John Morales, Natalie Nuno, Ekaterina Pavlovic, Wang-Chiew Tan, and Xiaolan Wang. 2019. Voyageur: An experiential travel search engine. In WWW, pages 3511–5. ACM.

Hugging Face. 2020. https://huggingface.co/distilbert-base-uncased-distilled-squad. Question Answering.

Cuiyun Gao, Jichuan Zeng, Michael R. Lyu, and Irwin King. 2018. Online app review analysis for identifying emerging issues. In ICSE, pages 48–58.

Ido Guy, Avihai Mejer, Alexander Nus, and Fiana Raiber. 2017a. Extracting and ranking travel tips from user-generated reviews. In WWW, pages 987–996.

Ido Guy, Avihai Mejer, Alexander Nus, and Fiana Raiber. 2017b. Extracting and ranking travel tips from user-generated reviews. In WWW, pages 987–996.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In WWW, pages 507–517.

Geoffrey E. Hinton and Sam T. Roweis. 2002. Stochastic neighbor embedding. In NIPS, pages 833–840.

Sharon Hirsch, Slava Novgorodov, Ido Guy, and Alexander Nus. 2021. Generating tips from product reviews. In WSDM, pages 310–318. ACM.

Xinyu Hua, Mitko Nikolov, Nikhil Badugu, and Lu Wang. 2019. Argument mining for understanding peer reviews. In NAACL-HLT, pages 2131–2137.

HuggingFace. 2020. https://huggingface.co/zero-shot/. Bart MNLI Zero Shot Topic Classification.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In ACL, pages 7871–7880.

Jinfeng Li, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. 2020. Deep or simple models for semantic tagging? it depends on your data. VLDB, 13(11):2549–2562.

Yuliang Li, Aaron Feng, Jinfeng Li, Saran Mumick, Alon Y. Halevy, Vivian Li, and Wang-Chiew Tan. 2019. Subjective databases. volume 12, pages 1330–1343.

Bing Liu. 2012. Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies.

LMScorer. 2018. Language model scorer. https://github.com/simonepri/lm-scorer.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In EMNLP, pages 404–411. ACL.

Samaneh Moghaddam. 2015. Beyond sentiment analysis: Mining defects and improvements from customer feedback. In ECIR, pages 400–410.

Sapna Negi and Paul Buitelaar. 2015. Towards the extraction of customer-to-customer suggestions from reviews. In EMNLP, pages 2159–2167.

Sapna Negi, Tobias Daudert, and Paul Buitelaar. 2019. Semeval-2019 task 9: Suggestion mining from online reviews and forums. In SemEval@NAACL-HLT, pages 877–887.

Slava Novgorodov, Guy Elad, Ido Guy, and Kira Radinsky. 2019. Generating product descriptions from user reviews. In WWW, pages 1354–1364.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Semantic Scholar.

TripAdvisor Hotel Reviews. 2021. https://www.cs.cmu.edu/~jiweil/html/hotel-review.html. In TripAdvisor.

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

Stefan Schweter and Alan Akbik. 2020. Flert: Document-level features for named entity recognition.

Shaohua Wang, NhatHai Phan, Yan Wang, and Yong Zhao. 2019. Extracting API tips from developer question and answer websites. In MSR, pages 321–332.

Xuan Zhang, Zhilei Qiao, Aman Ahuja, Weiguo Fan, Edward A. Fox, and Chandan K. Reddy. 2019. Discovering product defects and solutions from online user generated contents. In WWW, pages 3441–3447.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In ACL, pages 6236–6247.

Appendix

Non-informative Sentences Filtering

Recently large-scale pre-trained models (e.g. GPT-2 or BERT) have been used to select informative words. During pre-training, these models learned the informativeness of individual words in human vocabulary by reading massive natural texts such as Wikipedia articles. In GPT-2 (Radford et al., 2019), for example, the informative score of individual word in "There is a rooftop bar" is 0.00007, 0.26, 0.23, 0.00001, and 0.00037, respectively[4]. A lower score indicates more informativeness and adjectives (e.g. rooftop) or nouns (e.g. bar) usually show lower scores than stop words (e.g. is). With this favorable feature, pre-trained models can be used to rank sentences by their token informativeness.

Given pre-trained models can perceive the informativeness of individual words in a sentence, they can be used to filter out non-informative sentences. We use GPT-2, a representative pre-trained model, for the filtering. Specifically, for every sentence, we use GPT-2 to obtain informativeness scores of its tokens, then take product of scores. We sort all the sentences by the product scores and select top 20% sentences that have the highest scores. Since higher score means less informativeness, the top 20% sentences represent the most non-informative sentences and will be excluded from datasets.

To evaluate the effect of non-informative sentences filtering, we report the ratio of salient facts before and after filtering. Before filtering, the ratio is 16.3%, 6.8%, and 5.9% in Hotel, Product, and Restaurant dataset, respectively. After filtering, the ratio is 19.1%, 8.1%, and 6.1%, respectively. The ratio of salient facts increases in all of the three datasets. The results indicate that pre-trained models can effectively exclude non-informative sentences from datasets to boost the ratios of salient facts.

Implementation Details

In this section, we describe the training detail of the proposed model.

`Unique` Model. HuggingFace's implementation[5] of BERT is used for our `Unique` model to estimator the probability over review sentences. When fine-tuning the model, we use Adam (Kingma and Ba, 2014) as the optimizer with batch size of 64 and learning rate of 1e-5. The model is trained with the early stop mechanism where the training will end when there is no improvement on accuracy for three epochs. The model with the best accuracy is kept for testing. For each domain, we randomly sample ten entities for training, resulting in a total of training instances used for Hotel, Product, and Restaurant are 95,454, 44,560, and 356,505, respectively.

Irrelevance Classifier. Same as the `Unique` model, the HuggingFace's implementation of BERT is used for irrelevance classifier. As an ensemble model, a total of ten models is trained where each model is trained on nine-fold of data. The Adam (Kingma and Ba, 2014) is used as the optimizer with a batch size of 64 and a learning rate of 1e-6. The model is trained with the early stop mechanism where after 100 epochs, the training will end when there is no improvement on accuracy for five epochs. The model with the best accuracy is then kept. The overall ensemble model will take the average of the probabilities over all the ten models' predictions. Instances with averaged probability higher than 0.5 are classified as irrelevance and vice versa.

Data Annotation

We invite human editors to label sentences. Instructions for labeling are shown as follows. First, a salient fact sentence should be relevant to the targeted entity, i.e. mentioning at least one attribute/aspect of the entity. The purpose is to exclude irrelevant contents. Second, this attribute or aspect should be novel to readers. The purpose is to reveal unknown information of the entity to readers. Third, the salient fact sentence should use measurable descriptions. The purpose is to avoid subjective opinions that lead to biased understanding of the entity. We leave the understanding of the three conditions to annotators. We select the sentences that satisfy the first condition and meet either the second or third condition as salient facts.

To measure whether the labels are consistent, we randomly sample 100 sentences (i.e. 50 salient facts and 50 normal sentences) from the three domains. We invite two annotators to relabel these sentences and calculate cohen's kappa score as inter-annotator agreement. The score is 0.80 that is comparable to the results of existing helpful reviews annotation, e.g., 0.81 from a SEMEVAL-

[4] We obtain the scores using lm-scorer library (LMScorer, 2018) from Github

[5] https://github.com/huggingface/transformers

Domain	salient	normal
Hotel	complementary wine the rooftop deck Rooftop bar	an experience my stay a few small requests
Product	a thick liner note very computer savvy Win XP	Books this phone system well!2weeks
Restaurant	Ample parking $33.50 The 5 oz	bone marrow toast Excellent hashbrowns Customer service

Table 9: Key phrases extracted from salient facts and normal sentences, respectively. The comparison explains that salient facts reveal finer-grained attributes or quantitative descriptions of an entity that make them specific.

2019 Competition task 9 (Negi and Buitelaar, 2015) and 0.59 from TipRank (Guy et al., 2017b)). The result suggests that annotators have a high degree of agreement on salient facts.

Explanation of Saliency with Key Phrases

To understand what elements in a sentence make it salient, we extract key phrases of salient facts and normal sentences. We search span in a sentence that has the highest weights of BERT attention mechanism as key phrase. We present key phrase samples of salient spans and non-salient spans for Hotel, Product, and Restaurant domains in Table 9.

Salient facts show three patterns. First, the description targeted at attributes of an entity. In Product domain, for example, "thick liner note" or "Win XP" mention a specific product attribute, while "Books" and "well!2weeks" do not link to any attribute. Second, the attributes are novel that go beyond common knowledge. In Hotel domain, for example, "rooftop bar" or "wine" is unusual in hotel entities, compared with "an experience" and "small requests". Third, the description of an attribute reveals its quantity. In Restaurant domain, for example, "Ample parking" or "$33.50" relate to quantitative descriptions while "excellent hashbrowns" and "Customer service" do not reveal quantitative information of corresponding attributes. The results suggest that the most salient facts are those sentences that quantitatively describe novel attribute(s) of an entity.

Effect of `Unique` Model

We first evaluate the performance of `Unique` model that formulates salient fact extraction problem as entity prediction. Specifically, we randomly sample 10 entities and train a BERT model using reviews from the 10 entities. The total of training instances used for Hotel, Product, and Restaurant is 95,454, 44,560, and 356,505, respectively. The training takes a review to predict its targeted entity. After training, we compute the score for each review sentence using Equation 1. To evaluate the approach, we split data using 5-fold approach where one fold is used for finding the best threshold and the other four folds for testing. A total of five rounds are tested and F1 scores are averaged as the final score.

	Hotel	Product	Restaurant
Random guess	0.169	0.076	0.045
`Unique` model	0.395	0.205	0.114
w. Entity name removal	0.412	-	0.109
w. Irrelevance removal	0.401	0.201	0.128
ZL-Distiller	0.407	0.201	0.144

Table 10: Ablation study over Hotel, Product, and Restaurant datasets using ZL-Distiller. F1 of ZL-Distiller increases when turning on individual optimizations. Product has no entity name removal optimization because the dataset has no associated product names in the reviews.

We report F1 scores of `Unique` model on Hotel, Product, and Restaurant in Table 10. The F1 scores are 0.395, 0.205, and 0.114, respectively. To understand whether the F1 scores are significant, we evaluate the performance of random guess, a baseline that predicts a sentence as salient fact at the probability of $\%Positive$, with $\%Positives$ representing the ratio of positive labels of a dataset (see Table 3). The F1 score of random guess for Hotel, Product, and Restaurant is 0.153, 0.065, and 0.095, respectively and are lower than those of `Unique` model. The results suggest that `Unique` model can effectively improve extraction qualities of random guess, in various domains. Therefore, `Unique` is a strong signal of saliency that can be applied to different domains.

Effect of `Informative` Model

Effect of entity name removal. We evaluate the effect of entity name removal on `Unique` model and report F1 scores in Table 10, with exception for Product. Since the dataset has no associated product names in the reviews, we cannot enable the optimization. On Hotel, the F1 score of `Unique` model increases from 0.395 to 0.412, and the increment is 0.017. However, the F1 score on Restaurant decreases from 0.114 to 0.109. The

Model	Top-n			
	10	30	50	100
Hotel				
PacSum (tfidf)	**0.178**	0.308	0.376	0.370
ZL-Distiller +PacSum (tfidf)	0.133	**0.338**	**0.424**	**0.415**
Product				
PacSum (tfidf)	**0.414**	0.286	0.290	0.218
ZL-Distiller +PacSum (tfidf)	**0.414**	**0.367**	**0.319**	**0.252**
Restaurant				
PacSum (tfidf)	**0.200**	0.150	0.133	0.109
ZL-Distiller +PacSum (tfidf)	**0.200**	**0.300**	**0.200**	**0.145**

Table 11: Performance of zero-shot learning (i.e. PacSum and ZL-Distiller + PacSum) with zero labels. Best F1 scores are marked in bold.

decrement is 0.005. Overall, removing entity name does more good than harm. The results indicate that entity names overall mislead the Unique model and should be removed.

Effect of irrelevance removal. We evaluate the effect of irrelevant sentence removal on Unique model and report F1 scores in Table 10. After applying irrelevant sentences removal, the F1 score of Unique model on Hotel/Restaurant increases from 0.395/0.114 to 0.401/0.128. The increment is 0.006/0.014. However, the F1 score on Product decreases from 0.205 to 0.201, and the decrement is 0.004. Overall, the gain is higher than loss, so removing irrelevant sentences does more good than harm. The results indicate that irrelevant sentences overall mislead the Unique model and should be removed.

Overall effect. We evaluate the overall performance of ZL-Distiller when leveraging both Unique model and Informative model (i.e. turning on entity name removal and irrelevance removal simultaneously). We show F1 scores in Table 10. Compared with Unique model only, ZL-Distiller achieves 0.012 and 0.03 F1 gains on Hotel and Restaurant. Meanwhile, ZL-Distiller shows similar F1 on Product with a difference as small as 0.004. We anticipate ZL-Distiller can perform better on Product when entity names are present in the dataset.

Performance of Jointly Unsupervised Prediction

Since ZL-Distiller captures distributional patterns including "unique" and "informative", we would like to understand whether ZL-Distiller is still helpful to the state-of-the-art unsupervised extractor. For this purpose, we use PacSum (Zheng and Lapata, 2019), a recent extractive summarizer, as the representative unsupervised solution. We first obtain PacSum extraction performance using tfidf as sentence embedder. We next combine PacSum (tfidf) prediction scores with ZL-Distiller prediction scores and denote the combination as ZL-Distiller + PacSum (tfidf). Specifically, ZL-Distiller + PacsuM (tfidf) takes products of PacSum (tfidf) scores and ZL-Distiller scores then ranks sentences by product scores. We take the top-n as the predicted salient facts and vary *n* with 10, 30, 50, and 100. For each *n*, the F1 score is reported.

We present the F1 scores of PackSum (tfidf) and ZL-Distiller + PacSum (tfidf) in Table 11. ZL-Distiller + PacSum (tfidf) improves the F1 score of PacSum (tfidf) on 11 out of the 12 settings. Specifically, ZL-Distiller + PacSum (tfidf) outperforms PacSum (tfidf) on Product and Restaurant on all of the top 10, 30, 50, and 100 settings, and on Hotel on top 30, 50, and 100 settings. The results suggest that ZL-Distiller overall is helpful to the state-of-the-art unsupervised solution towards better extraction performance.

Technical Novelty

Herein, we proposed to exploit distributional patterns for review mining tasks. Our results demonstrate that distributional patterns are auxiliary patches to salient fact extraction as they lead to better performance when combined together. Therefore, we expect that the deployment of distributional patterns in relevant studies, such as helpful review prediction or suggestion mining, can also generate better results, which will extensively expand the applications of our proposed pattern in the field of review mining. We also proposed a scoring mechanism that works well on a variety of domains (i.e., hotel, product, restaurant) in both supervised and unsupervised settings. The scoring mechanism together with target_name and irrelevant_sentence_removal models lead to unbiased and unique results in Question Answering and Entity Summarization, compared to the results without their processing. Finding useful reviews is of high practical importance and can be applied to many NLP problems. We chose the most appropriate mechanisms instead of developing new methods to have the best results. In the future, we will apply this task to mining more informative reviews for a variety of NLP domains and applications.