Research Presentation

-Zishan Ahmad
IIT Patna

Outline

1. Active Learning based Relation Classification for Knowledge Graph Construction from Conversation Data
2. Tweet to News Conversion: An Investigation into Unsupervised Controllable Text Generation
3. Unsupervised aspect-level sentiment controllable style transfer

Active Learning based Relation Classification for Knowledge Graph Construction from Conversation Data

Presented By: Zishan Ahmad

Authors: Zishan Ahmad, Asif Ekbal, Shubhashis Sengupta, Anutosh Mitra, Roshni Rammani, and Pushpak Bhattacharyya

Ahmad, Zishan, et al. "Active Learning Based Relation Classification for Knowledge Graph Construction from Conversation Data." *International Conference on Neural Information Processing*. Springer, Cham, 2020.

Overview

- The knowledge graph represents a collection of interlinked descriptions of entities – objects, events or concepts
- Knowledge graphs put data in context via linking and semantic metadata
- They provide a framework for data integration, unification, analytics and sharing
- **Eg:** Wikidata, Never-Ending Language Learner (NELL), YAGO

Overview

- A conversational Knowledge Graph should store the entities and relation between them in terms of the conversational context

- The KG should update these relation and entities according to the conversation as the conversation proceeds:

    - The relations should be negated according to the conversation
    - The entities should be updated according to the conversation

Overview

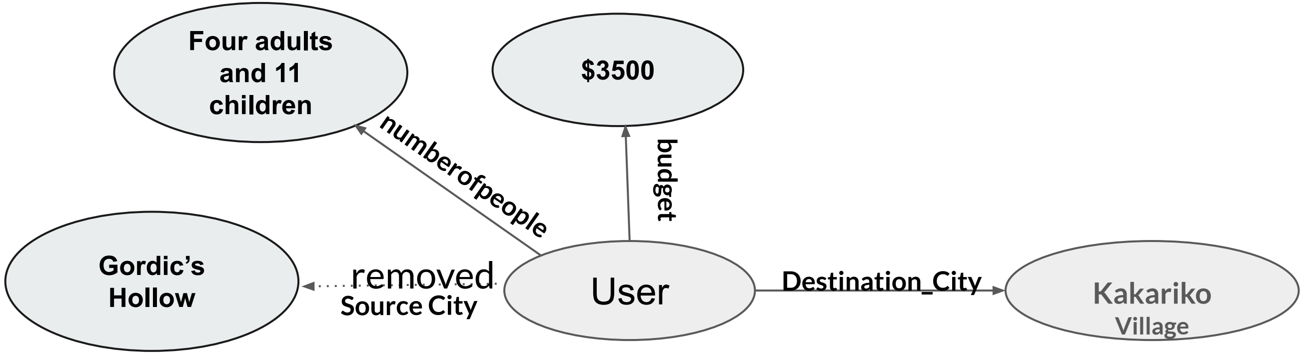
- Given a conversation between a user and an agent, the task is to extract entity-relation triples, classify the triple relation to ontological relation types, and then construct a KG for the conversation.
- An example knowledge graph construction is given below:
  - **User**: Hi, me and my family want to take a trip to Kakariko Village.
  - **Agent**: How many adults and how many children will you be bringing with you? Do you have a preferred departure location? What would be your maximum budget?

Knowledge Graph

Overview

User: We are four adults and 11 children departing from Godric's Hollow. $3500 is the most we are willing to spend.

Agent: I do not have any packages available from that departure location for that budget.



Knowledge Graph

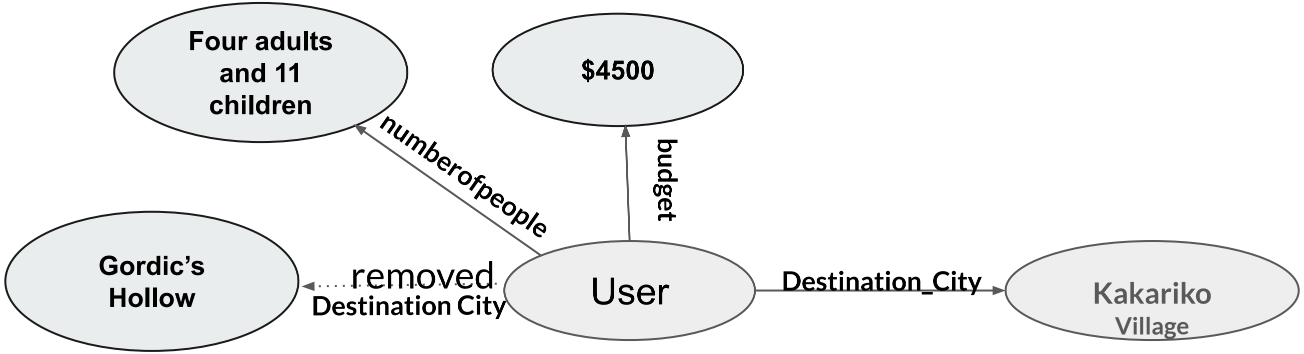
User: I would not be able to spend more than $4500

Agent: Would you like to suggest another departure location?

Overview

User: I would not be able to spend more than $4500

Agent: Would you like to suggest another departure location?

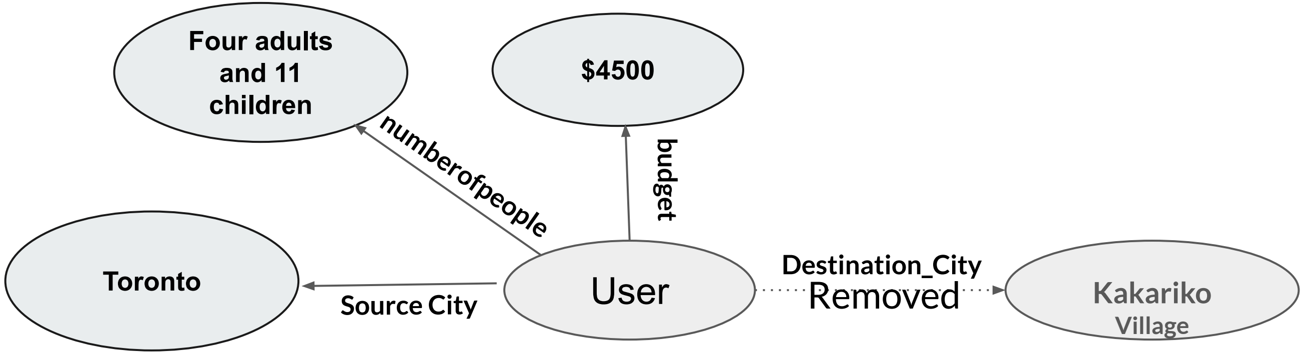


Knowledge Graph

Overview

User: We'll catch a flight from Toronto then. We're headed for Kakariko Village.

Agent: There are no travel packages available to Kakariko Village.



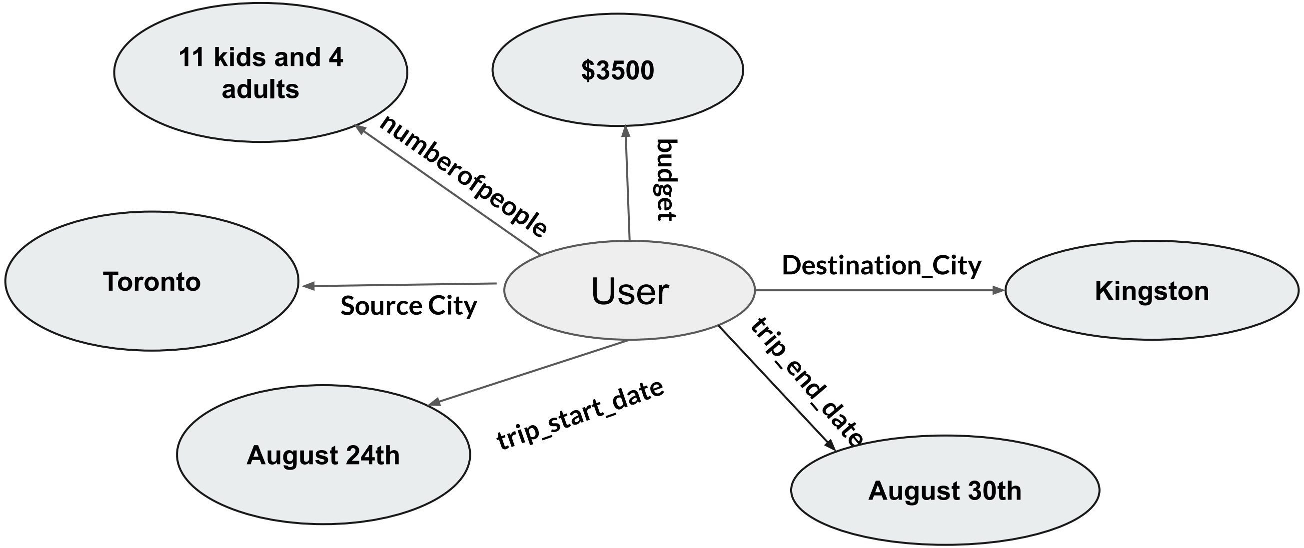
Knowledge Graph

Overview

User: Ok, do you have any travel packages to Kingston, Canada for 11 kids and 4 adults, $3500 or less, departing from Toronto?

Agent: Yes, I have a package available leaving Toronto on August 24th and returning to Toronto on August 30th.

User: Yes, I'll take it. Thank you



Knowledge Graph

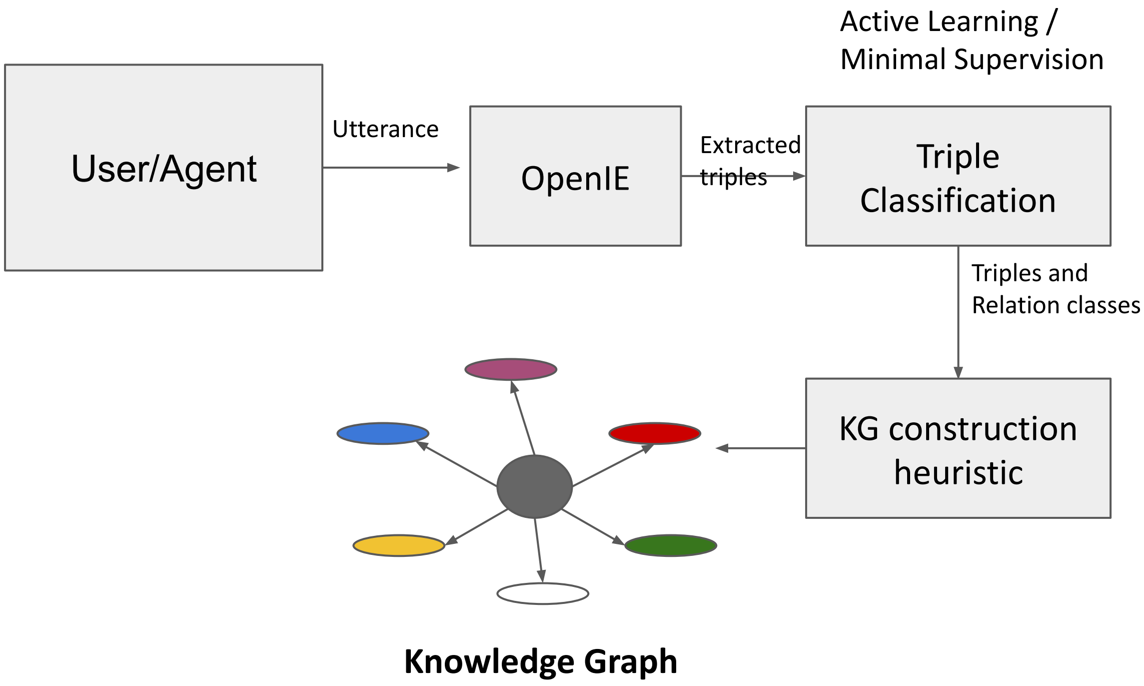
Overview

- **Challenges**
  - The beliefs in conversation data are not grounded and can change with dialogue turns
  - The relation between entities can be implicit in nature

- **Motivation**
  - Using knowledge graph for generating conversations can potentially help in grounding the conversation and making it coherent
  - In a conversation between a user and a call center agent (e.g. Travel Agency), the system could extract entities and relationships at each turn, and a final KG of these relation could be constructed by the end of the conversation. This final KG represents the final user requirements, (Travel Booking details etc.) and queries could be executed to satisfy these requirements automatically (Query for ticket bookings etc)
  - Augmenting Knowledge Graphs in neural dialogue system can help tackle the issue of data sparsity

Overview

Proposed System



Dataset Description

Datasets used:

- MS Frames
- MS e2e Taxi

Dataset	Total Dialogues	Total Dialogue Turns	Avg Turns per Dialogue
MS Frames	1,369	19,986	15
MS e2e taxi	3,094	23,312	7

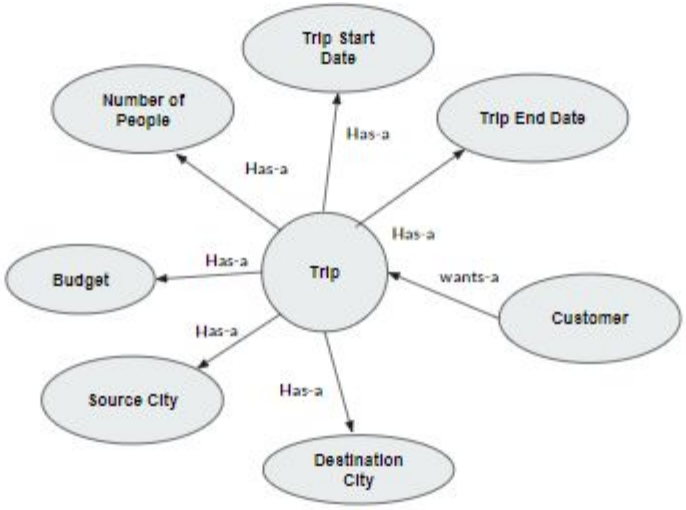
Dataset Description

Class distribution after active learning

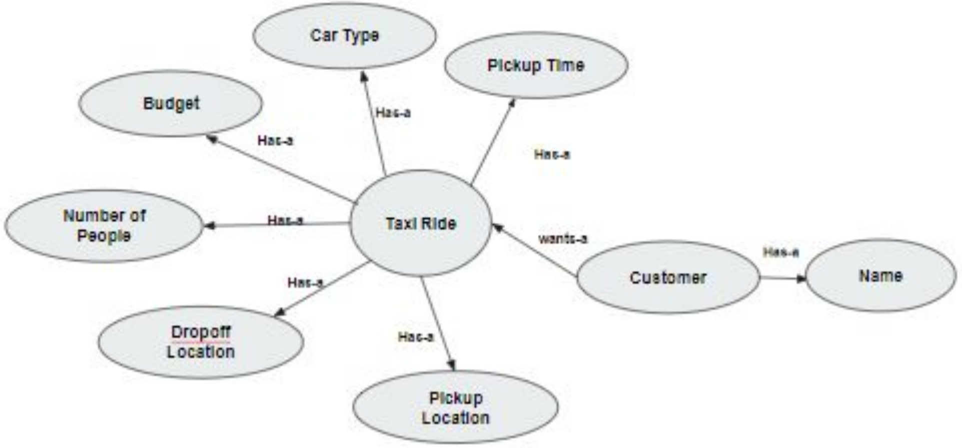
MS Frames		MS e2e Taxi	
Relation Class	**# Samples**	**Relation Class**	**# Samples**
Source_City	148	Pickup_Location	112
Destination City	131	Dropoff_Loaction	110
Trip_Start_Date	102	Pickup_Time	74
Trip_End_Date	81	Car_Type	115
Number_of_People	77	Number_of_People	120
Budget	101	Budget	13
Other	360	Customer_Name	75
		Other	381

Methodology

- Following ontology was created for Knowledge Graphs of the datasets:



Ontology for Microsoft Frames Dataset



Ontology for Microsoft e2e taxi Dataset

Methodology

Relation classification algorithm:

Trained BERT based classifier for relation classification using the following steps:

1. Enrich the sentence with the extracted OpenIE triple. Eg:
   - **Statement:** I need to go to Denver Airport.
   - **OpenIE triple extracted:** {'subject': 'I', 'relation': 'go to', 'object': 'Denver Airport'}
   - **Enriched Statement:** <tok1>I</tok1> need to go to <tok2>Denver Airport</tok2>.
   - **Relation Class:** Dropoff_Location
2. Tokenize the enriched sentence using BERT tokenizer and assign them their token indices
3. Pass the indexed sentence to the BERT model and extract the representation of CLS token at output layer
4. Pass the CLS representation to a linear layer and apply Softmax for classification

Methodology

Active Learning Steps for Relation classification:

1. Train the classifier on seed data.
2. Run the trained classifier on un-annotated data.
3. Obtain the final class prediction and probabilities of prediction using the final Softmax layer of the classifier.
4. Calculate entropy of prediction probabilities using the following formula:

   $$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

   Here, H(X) is the entropy w.r. ber of classes, p_i is the probability of prediction of the i^{th} class.

5. From the annotated data obtained from classifier, add the samples with top k largest entropy to the seed sample.

Methodology

Active Learning Steps for Relation classification (continued):

5. if accuracy of the classifier increases by $\gamma\%$ and seed sample is less than α
6. go-to step 1
7. else stop and save the final trained classifier

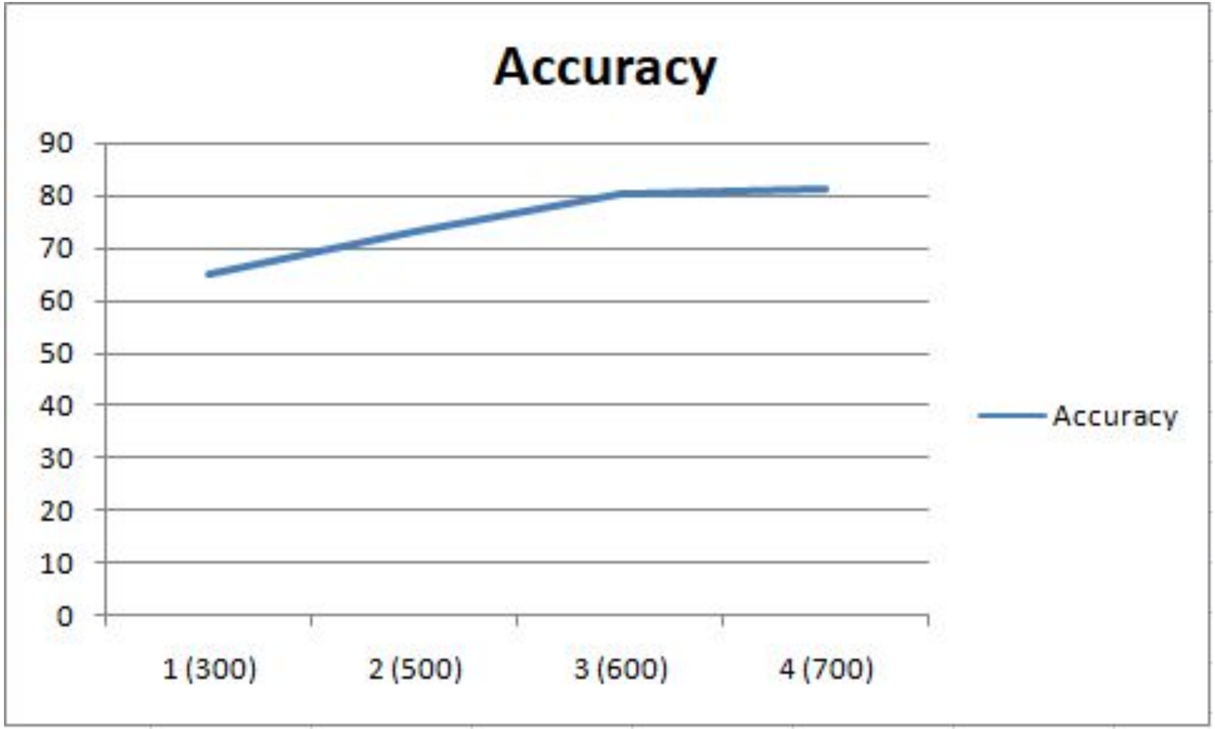
Methodology

Algorithm for Knowledge Graph construction:

1. Initialize KG (Knowledge Graph)
2. SRL ← Semantic Role Labeller
3. C ← Pre-trained Relation Classifier
4. for utterance in conversation do:
    a. split utterance into sentences
    b. for sentence in sentences do:
        i. OIE ← Get all the OpenIE triples from the sentence
        ii. C_Trip ← C(OIE)
        iii. Add the C_Trip triples to the corresponding KG slot
        iv. SRL_Sent ← SRL(sentence)
        v. If negation exists in SRL_Sent then
        vi. Arg ← Negated Argument
        vii. Remove all triples from KG containing Arg
5. Return KG

Results

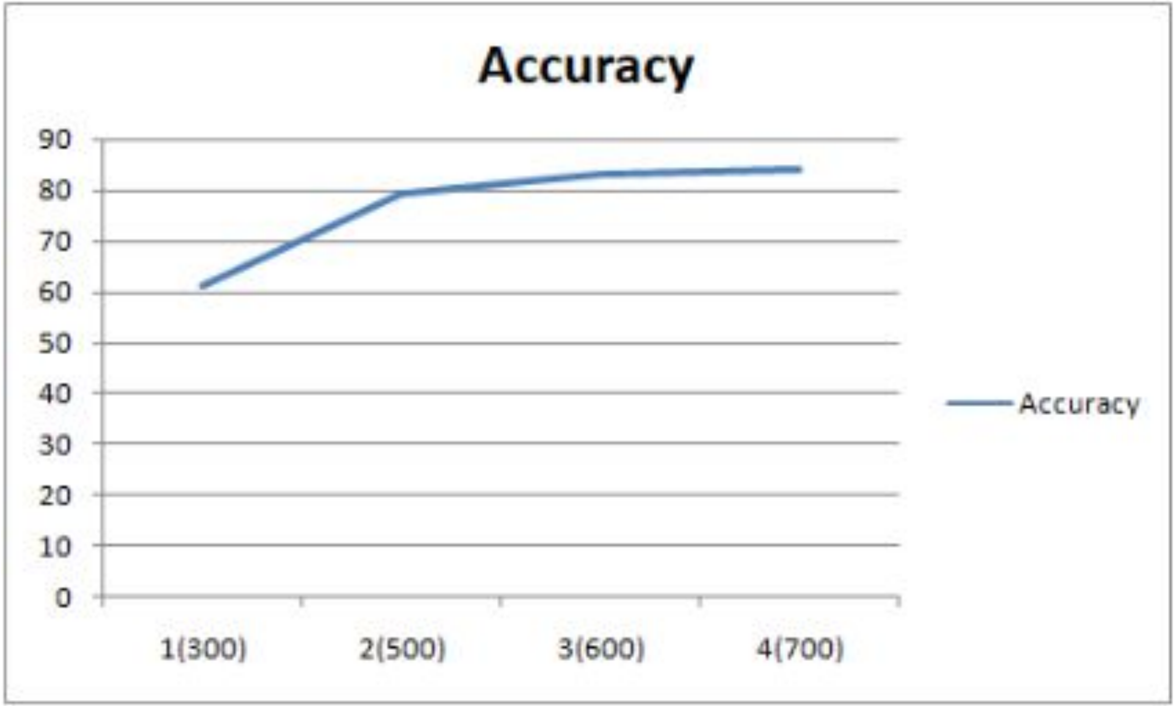
Active Learning accuracy v/s steps (MS Frames Dataset):



X axis: Active learning step (number of training sample); Y-axis: Accuracy of the classifier

Results

Active Learning accuracy v/s steps (MS e2e taxi booking dataset):



X axis: Active learning step (number of training sample); Y-axis: Accuracy of the classifier

Results

Relation Classification:

Microsoft Frames Classifier				Microsoft e2e Classifier			
Class	**Precision**	**Recall**	**F1-Score**	**Class**	**Precision**	**Recall**	**F1-Score**
Source_City	0.88	0.90	0.89	*Pickup_Location*	0.80	0.61	0.69
Destination_City	0.85	0.90	0.88	*Dropoff_Location*	0.74	0.89	0.81
Trip_Start_Date	0.78	0.68	0.72	*Pickup_Time*	0.89	0.89	0.89
Trip_End_Date	0.83	0.74	0.79	*Car_Type*	0.69	0.76	0.72
Number_of_People	0.81	0.79	0.80	*Number_of_People*	0.95	0.91	0.93
Budget	0.78	0.80	0.89	*Budget*	1.0	0.75	0.86
Other	0.83	0.76	0.80	*Customer_Name*	0.90	0.94	0.92
				Other	0.83	0.84	0.84

Results

Knowledge Graph Construction:

Dataset	Correctness	Completeness
Microsoft Frames	64%	56%
Microsoft e2e	71%	40%

Conclusion

- First attempt at building a KG for conversational data
- The experiments show that we can obtain good results with minimal supervision (as few as 700 samples were used for training)
- The results can further be improved by using better negation handling techniques and better triple extraction

Tweet to News Conversion: An Investigation into Unsupervised Controllable Text Generation[1]

Zishan Ahmad, Mukuntha N S, Asif Ekbal and Pushpak Bhattacharyya

Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, India
Samsung Research India, Bangalore, India

[1] Ahmad, Zishan, et al. "Tweet to News Conversion: An Investigation into Unsupervised Controllable Text Generation." 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020.

Problem Definition

- Our task is to formulate a news paragraph from a set of tweets about a disaster event
- The generated news paragraph should contains all the information provided in the tweets, without producing any extra or repetitive information

- **Input:**
  - **Tweet 1 (Input 1):** breaking: at least 126 killed in taliban attack on pakistani school.
  - **Tweet 2 (Input 2):** update: #peshawarattack over, all hostage-takers dead - police.
  - **Tweet 3 (Input 3):** dozens of children killed as taliban gunmen storm peshawar school.
  - **Tweet 4 (Input 4):** #pakistan school attack over, all six attackers are dead. #peshawarattack #talibanattacksschool
- **News (Output):** Taliban gunmen had stormed a school in Pakistan, killing at least 126. Police stated that the Peshawar attack is over as all six hostage-takers are dead

Contribution

- A new method to convert informal and noisy tweet content into a more formal news-text format.
- A novel training regime to stitch together the information nuggets into long coherent sentences.
- An evaluation dataset for the task, consisting of 1265 instances of a set of four different domains of disasters and their corresponding news paragraphs.

Dataset

- Our dataset comprises of a collection parallel tweet-news created by us
- For the purpose of style transfer training we use non parallel set of tweets and news
- We break news data into clauses and obtain parallel clause-news pairs

Dataset	Number of Instances
News-Tweet-Parallel Validation	265
News-Tweet-Parallel Test	1,000
Tweet-Non-Parallel	45,295
WMT-Non-Parallel	171,400
News-Clause-Pair	126,120

Methodology (1/6)

The entire system for creating news paragraph from tweets is built in two parts:

- **Part 1:** This system converts the style of individual tweets into more formal news style

- **Part 2:** This system takes as input four tweets converted to news sentences (from Part 1) and stitches them together to form a coherent news paragraph

Methodology (2/6)

Part 1:

- We take two non-parallel datasets of two different styles:
  - $X = \{x_1, x_2, \ldots x_m\}$ consisting of tweets
  - $Y = \{y_1, y_2, \ldots y_n\}$ consisting of news sentences.
- We use pre-trained XLM encoder-decoder model for our task
- For training we use unsupervised neural machine translation (UNMT) steps to achieve style transfer between tweets and news

Methodology (3/6)

Part 1 (UNMT steps)

- **Denoising Auto-Encoding:**
  - In this step noise in the form of random masking, shuffling and dropping is introduced to the input of the XLM model
  - The decoder is trained to reconstruct the original de-noised sentence
  - This is done in order to make the model learn the two style distributions
- **On-the-fly-back-translation**:
  - We use tweets (T) as input and set the style embedding as tweets at encoder
  - At decoder we set the style embedding to news and obtain synthetic news (N')
  - We pass this synthetic news N' at input to encoder and set the encoder embedding to news
  - Setting the tweet embeddings at decoder we obtain T'. and compute the cross-entropy loss with original T
  - We repeat the same steps with news (N) as input.

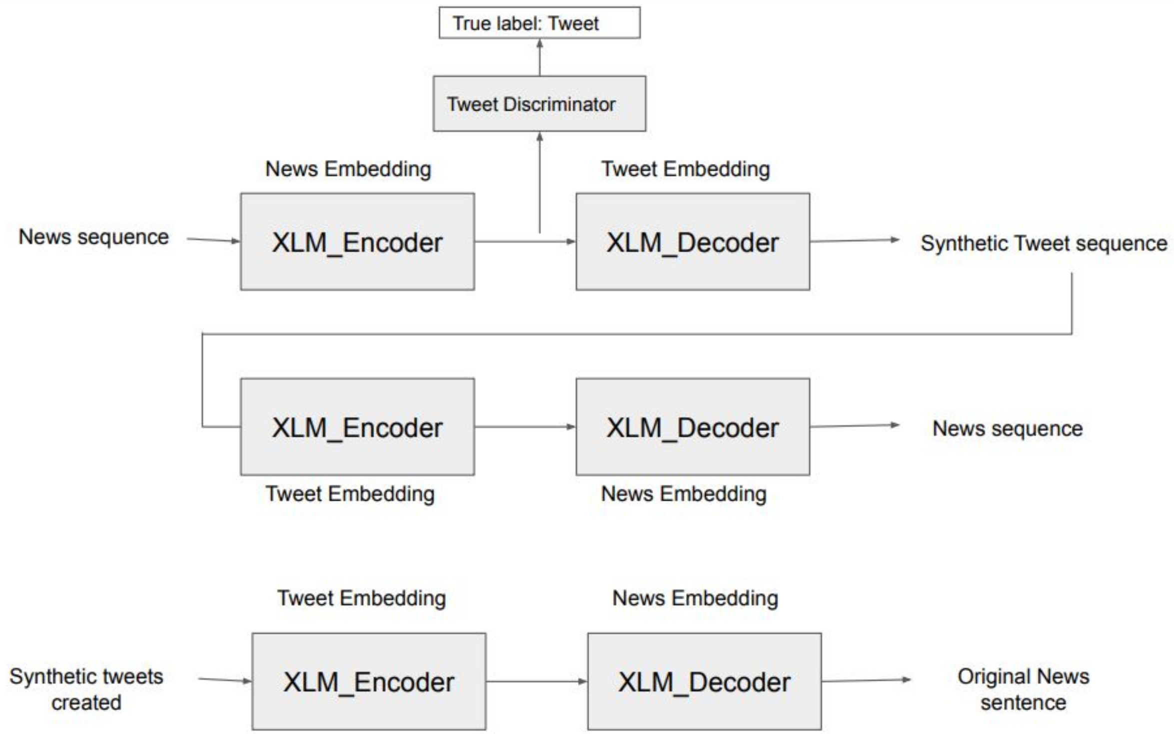
Methodology (4/6)

Part 1

- **Adversarial Training (DIS):**
  - We additionally train a Gated Recurrent Unit (GRU) based discriminator
  - It takes in the content vectors $z_x = E(x, l_1)$ and $z_y = E(y, l_2)$ produced by the encoder and classify them as tweet or news
  - The discriminator is trained as a binary classifier that outputs the probability that a given latent content vector z comes from a tweet or news
  - The adversarial loss L_D used to train the discriminator weights and the encoder is trained with the reverse objective $-L_D$
- **Synthetic parallel-data training step (SYN)**:
  - Paraphrase the English news sentence by using pre-trained translation modules to translate it to a distant language and translating it back to English.
  - Introduce random spelling mistakes and inject random hashtags to the news sentence
  - This synthetic data is used to train the same encoder-decoder model as a separate step

Methodology (5/6)

Part 1:



Methodology (6/6)

Part-2

- **XLM-Merge:**
  - We use ClauseIE[1] to split the news sentences (N) into clauses
  - We concatenate the clauses and obtain input sequence (CI)
  - We train another XLM model by using the CI as input and N at output
  - This way the model is trained to stitch together multiple sentences into a single sentence
  - During inference the model takes tweets converted to news 2 at a time and stitched them together

[1] http://resources.mpi-inf.mpg.de/d5/clausie/clausie-www13.pdf

Evaluation Metrics

- **Automatic Evaluation:**
  - **BLEU Score:** BLEU measures the n-gram overlap between a generated response and a gold response

- **Manual Evaluation:**
  - **Fluency (1-5):** It measures the grammatical correctness of the predicted response
  - **Adequacy (1-5):** It measures whether the generated sequence is semantically the same as the gold sequence

Results:

	BLEU	Adequacy	Fluency
XLM-UNMT	14.34	2.206	2.245
XLM-UNMT + SYN	18.28	**3.52**	2.34
XLM-Merge	19.04	2.72	2.63
XLM-UNMT+Disc+XLM-Merge	16.31	2.74	2.7
XLM-UNMT + SYN + XLM-MERGE	19.04	3.04	3.02
XLM-UNMT+Disc + SYN + XLM-MERGE	**19.32**	3.04	**3.2**

Analysis

- Our model tries to join sentences even when they are very different and not suitable for joining.

  Input:

  Tweet1: rain brings relief to dhaka as cyclone #faniapproaches .

  Tweet2: two people are reported to have died and more than a million have fled their homes after #cyclonefani made landfall on india's east coast. more on this story here:.

  Tweet3: thousands evacuated in eastern india as cyclone fani approaches.

  Tweet4: 'extremely severe' cyclone fani to hit south of puri on friday: ndma.

  Output: rain brings relief to dhaka as two people are reported to have died and more than a million have fled their homes after cyclone fani made landfall on india's east coast. more on this story in eastern india as cyclone approaches , extremely severe .

Unsupervised Aspect-Level Sentiment Controllable Style Transfer[1]

Zishan Ahmad, Mukuntha N S, Asif Ekbal, Pushpak Bhattacharyya

Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, India

[1]Sundararaman, M. N., Ahmad, Z., Ekbal, A., & Bhattacharyya, P. (2020, December). Unsupervised Aspect-Level Sentiment Controllable Style Transfer. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing (pp. 303-312).

Literature Survey

- **Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R. and Black, A.W. Style Transfer Through Back-Translation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 866-876).**
  - Proposed style transfer task with respect to sentiment, gender and political slant
  - Used back-translation method based on Long-Short-Term-Memory (LSTM) encoder-decoder models coupled with style discriminators

- **Li, J., Jia, R., He, H., & Liang, P. Delete, Retrieve, Generate: a Simple Approach to Sentiment and Style Transfer. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Volume 1 : Long Papers) (pp. 1865-1874).**
  - Proposed style transfer for sentiment
  - Used method to delete and retrieve phrases at input and show that properly deleting and retrieving phrases helps in style transfer

Aspect Based Sentiment Analysis

- Aspect-based sentiment analysis (ABSA) is a text analysis technique that breaks down text into aspects (attributes or components of a product or service), and then allocates each one a sentiment level (positive, negative or neutral)

- **Eg:**
  - The **service** was speedy and the **salads** were great, but the **chicken** was bland and stale.

Aspect	Sentiment
service	positive
salads	positive
chicken	negative

Problem Definition

- Controlling sentiment for any aspect in the given text
- **Given a set of labelled sentences**
    - $D = \{(x_1; l_1) : : : (x_n; l_n)\}$, where x_i is a sentence, and $l_i = \{(t_{i1}; p_{i1}) : : : (t_{im}; p_{im})\}$
    - Where : t_{ij} is aspect-target , p_{ij} is sentiment-polarity expressed towards t_{ij} and p_{ij} ε {positive; negative}
- A model is to be learned that takes as input $(x; l_{tgt})$ where x is the source sentence expressing some aspect polarity set l_{src}, and outputs y that retains all the non-polarity content in x while expressing the aspect-polarity set l_{tgt}
- This model is to be learnt without any parallel data

Problem Definition (cont..)

The *service* was speedy and the *salads* were great, but the *chicken* was bland and stale.

Service - Positive
Salads - Positive
Chicken - Negative

Query:
Service - Negative
Salads - Positive
Chicken - Positive

The *service* was slow, but the *salads* were great and the *chicken* was tasty and fresh.

Service - Negative
Salads - Positive
Chicken - Positive

Challenges

- Lack of parallel data where input and output sentences are aligned according to the input query
- A single piece of text (such as a single sentence) can express a positive sentiment towards an aspect-term and negative sentiment towards another aspect term
- The style to be controlled is more localized and fine grained thus making the task more challenging
- Any number of aspect-terms can be present in the text and the model should be able to control sentiment even for previously unseen aspect terms

Key contributions

- We propose a new task of controlling aspect level sentiment in a given text
- We propose a novel encoder-decoder based architecture is proposed to perform unsupervised aspect-level sentiment transfer.
- We propose a novel saliency based 'polarity injection' method to the hidden representation of the encoder to complete the query for the decoder

Dataset

- A large dataset with aspects-terms and their polarities is required
- We train a BERT aspect term extractor and aspect polarity classification based on Xu et al. (2019) on SemEval task.
- Using the trained BERT models we annotate Yelp reviews dataset for aspects and their polarities
- Our style transfer model is trained on this silver label Yelp reviews dataset

Dataset Statistics

Dataset	No. of Sentences	No. of Target Aspects	No. of Unique Target Aspects
SemEval (Train and Validation)	2,242	4,016	1,437
SemEval (Test)	401	513	269
Yelp (Train and Validation)	361,968	471,820	47,750

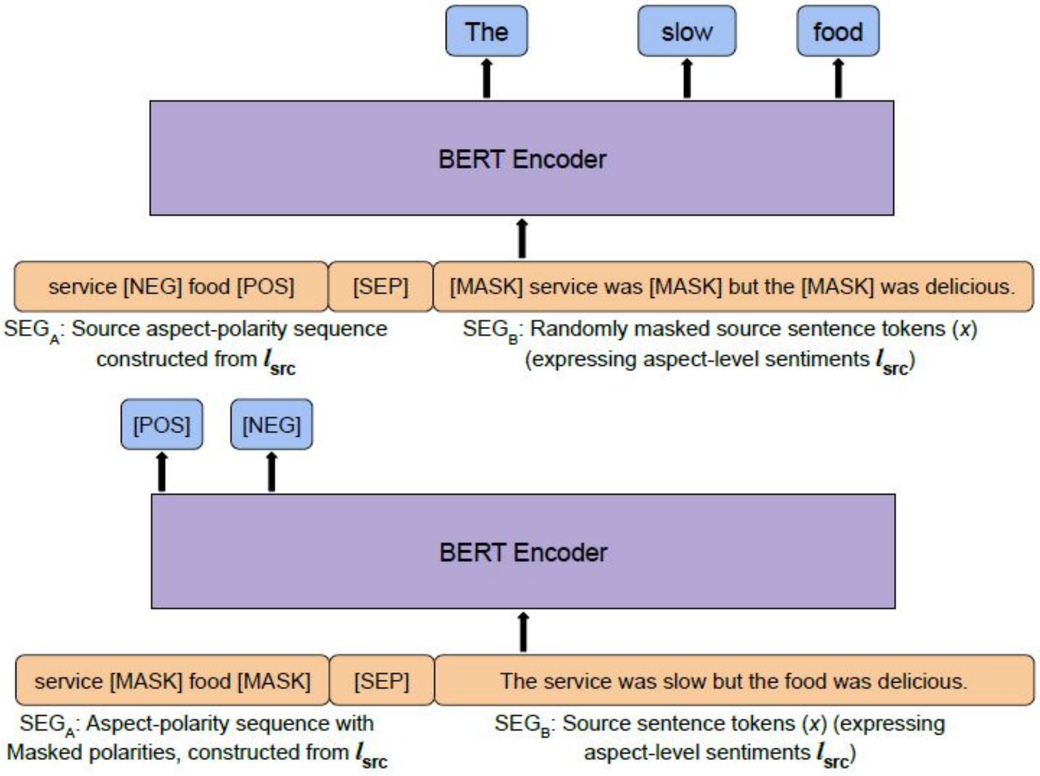
Methodology (⅛)

- **Input Representation**
  - The input to the model is the a pair of sequence separated by a [SEP] token. ($[SEG_A][SEP][SEG_B]$)
  - SEG_A = "$T_1P_1[SEP_{ASP}] : : : T_kP_k[SEP_{ASP}]$" where T_i is the tokenized target aspect term and P_i ε {[POS]; [NEG]} is a polarity corresponding to it. $[SEP_{ASP}]$ is a separator token.
  - SEG_B consists of a sentence expressing some sentiment towards these targets.

Methodology (2/8)

- **Preconditioning the BERT-encoder for ABSA Input Representations**
  - We precondition the BERT encoder to better understand the ABSA task and to learn the token embeddings for [POS] and [NEG] with MLM pre-training (a cloze objective)
  - For each data instance, with an equal probability, we randomly mask out either
    - All the polarity tokens from aspect-polarity sequence (SEG_A)
    - Random tokens from the sentence (SEG_B)
  - Train the encoder to correctly predict the masked-out tokens

Methodology (⅜)



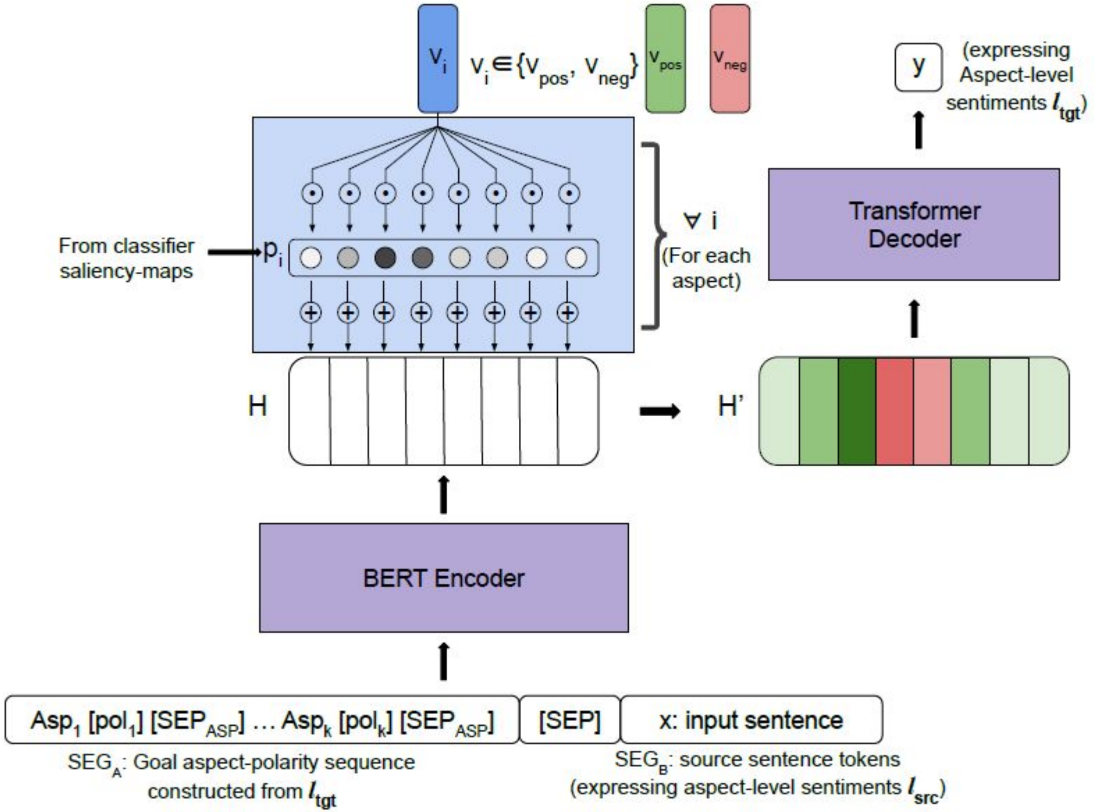
Methodology (4/8)

- **Encoder Decoder Architecture:**
  - The architecture is a Transformer model with an encoder and a decoder
  - BERT is used at the encoder of the model
  - The input consists of target polarities in [SEG_A] and the source sentence whose aspect polarities are to be transformed in [SEG_B]
  - The polarity-injection adds the weighted target polarities l_{tgt} into the hidden-representation H from the BERT encoder, to obtain H_0 which is passed to the decoder

Methodology (⅝)

- **Polarity Injection**
  - We inject (add) weighted amounts of two vectors corresponding to positive and negative sentiments to edit the hidden states output by the encoder
  - For each aspect, the vector added corresponds to the desired target polarity of this aspect
  - The amount of polarity added to a given token depends on the saliency-based weight for this token calculated from the gradient for this aspect's polarity from the ABSA polarity classification model
  - The following equation is used to update the hidden states:
    - $h'_j = h_j + \sum p_{ij}.v_i$
    - $v_i = v_{pos}$ if pol_i desired is positive and $v_i = v_{neg}$ if pol_i desired is negative
      p_{ij} is the saliency w.r.t polarity i and for token j

Methodology (6/8)



Methodology (7/8)

- **One-Zero Alternative to Saliency**

    - To test for performance in the absence of any saliency information, we also propose using a one-zero setup
    - Here p_{ij} is set to 1 over the tokens corresponding to the i_{th} aspect-term and 0 elsewhere
    - In this setup, v_{pos} gets added to the tokens corresponding to the positive aspects and v_{neg} gets added to the tokens corresponding to the negative aspects.

Methodology (8/8)

- **Unsupervised Training**
    - We alternate training steps between a denoising auto-encoding objective and a back-translation objective
    - During the denoising step, we add random noise to the sentence part of the input SEG_B
    - We also randomly mask the polarities in the aspect polarity sequence in SEG_A with a small probability to ensure the model learns to generate outputs using the polarity injection clues
    - During the backtranslation step, a random query ltgt aspect-polarity sequence is used to produce an intermediate translation (using the model)
    - The same model is trained to regenerate the original input when provided the aspect-polarity sequence from the original input sentence

Experiments

- **We conduct four experiments:**
  - **BERT-Baseline (BB):** In this experiment we do not use any polarity injection or MLM step
  - **BB + MLM pre-training (BBMLM):** We perform MLM pre-training before the back-translation step
  - **BB-MLM + one-zero polarity injection:** We perform MLM pre-training before the back-translation step and we also use one-zero polarity injection
  - **BB-MLM + saliency-based polarity injection:** We perform MLM pre-training before the back-translation step and we also use saliency-based polarity injection

Evaluation Metrics

- **Automatic Evaluations:**
  - **BLEU Score:** Human reference outputs were written for 100 of the queries, which are used to compute BLEU scores
  - **Classifier Score:** The fraction of aspect-level sentiment polarities (predicted by the classifier from the output) that match with the desired aspect-level polarity (from the query)

- **Manual Evaluation:**
  - Human experts were asked to rate the output sentences Likert-scale (1 to 5) for three criteria:
    - **Aspect-level polarities (Att):** Check if the polarities in the generated text match the query set of aspect-level polarities
    - **Fluency (Gra):** Measuring the naturalness of the output
    - **Content preservation (Con):** Measuring the non polarity information in preservation

Results (Automatic Evaluation)

Model	Classifier Score (Overall)	Classifier Score (1-Aspect)	Classifier Score (2-Aspects)	Classifier Score (3-or-more Aspects)	BLEU Score
BERT-Baseline (BB)	0.5158	0.4983	0.5448	0.5036	36.0683
BB + MLM pre-training (BBMLM)	0.5298	0.5433	0.5310	0.5145	35.4601
BB-MLM + one-zero polarity injection	0.5415	0.5675	0.5276	0.5290	35.8244
BB-MLM + saliency-based polarity injection	**0.5918**	**0.6125**	**0.5828**	**0.5797**	**39.3838**

Results (Manual Evaluation)

Model	Att	Con	Gra
BERT-Baseline (BB)	2.48	3.99	3.96
BB + MLM pre-training (BBMLM)	2.64	3.95	4.04
BB-MLM + one-zero polarity injection	2.80	4.00	**4.05**
BB-MLM + saliency-based polarity injection	**2.98**	**4.08**	**4.05**

Outputs

Input	Query	Model Output
overall, decent food at a good price, with friendly people.	food - negative people - positive	overall , mediocre food at a good price , with friendly people .
	food - positive people - negative	overall , decent food at a good price , with rude people .
the waiter was attentive , the food was delicious and the views of the city were great	waiter - negative food - positive views of the city - positive	the waiter was inattentive , the food was delicious and the views of the city were great .
	waiter - positive food - negative views of the city - positive	the waiter was attentive , the food was disappointing

Error Analysis (1/2)

Input	i’d be horrified if my staff were turning away customers so early and so rudely!
Query	staff - positive
Output	i’d be delighted if my staff were turning away customers so early and nicely!
Comment	Lower naturalness of the output from real-world knowledge that turning away customers is bad.

Error Analysis (2/3)

Input	i must say i am surprised by the bad reviews of the restaurant earlier in the year , though .
Query	restaurant - negative
Output	i must say i am surprised by the bad reviews of the restaurant earlier in the year , though .
Comment	No change. Sentiment here is implied and latent.

Thank You!