6 Top NLP Papers

These papers provide a breadth of information about NLP (***Natural language processing*** – a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human languages, in particular how to program computers to process and analyze large amounts of natural language data) that is generally useful and interesting from a computer science perspective.

Contents

Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems

Chien-Sheng Wu[†]*, **Andrea Madotto**[†], **Ehsan Hosseini-Asl**[‡], **Caiming Xiong**[‡], **Richard Socher**[‡] **and Pascale Fung**[†]
[†]The Hong Kong University of Science and Technology
[‡]Salesforce Research
jason.wu@connect.ust.hk

Abstract

Over-dependence on domain ontology and lack of knowledge sharing across domains are two practical and yet less studied problems of dialogue state tracking. Existing approaches generally fall short in tracking unknown slot values during inference and often have difficulties in adapting to new domains. In this paper, we propose a **TRA**nsferable **D**ialogue stat**E** generator (**TRADE**) that generates dialogue states from utterances using a copy mechanism, facilitating knowledge transfer when predicting *(domain, slot, value)* triplets not encountered during training. Our model is composed of an utterance encoder, a slot gate, and a state generator, which are shared across domains. Empirical results demonstrate that TRADE achieves state-of-the-art joint goal accuracy of 48.62% for the five domains of MultiWOZ, a human-human dialogue dataset. In addition, we show its transferring ability by simulating zero-shot and few-shot dialogue state tracking for unseen domains. TRADE achieves 60.58% joint goal accuracy in one of the zero-shot domains, and is able to adapt to few-shot cases without forgetting already trained domains.

1 Introduction

Dialogue state tracking (DST) is a core component in task-oriented dialogue systems, such as restaurant reservation or ticket booking. The goal of DST is to extract user goals/intentions expressed during conversation and to encode them as a compact set of dialogue states, i.e., a set of slots and their corresponding values. For example, as shown in Fig. 1, *(slot, value)* pairs such as *(price, cheap)* and *(area, centre)* are extracted from the conversation. Accurate DST performance is crucial for appropriate dialogue management, where user intention determines the next system action and/or the content to query from the databases.

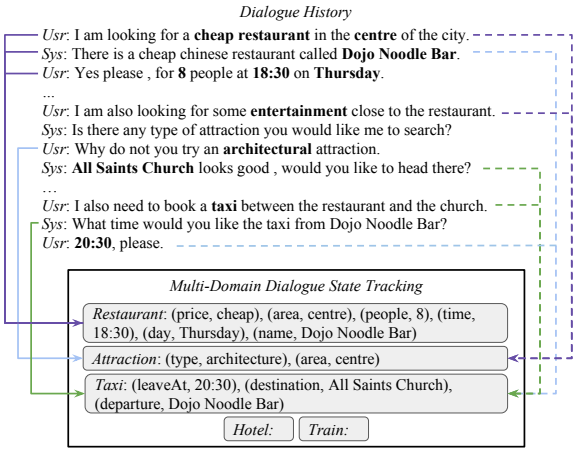


Figure 1: An example of multi-domain dialogue state tracking in a conversation. The solid arrows on the left are the single-turn mapping, and the dot arrows on the right are multi-turn mapping. The state tracker needs to track slot values mentioned by the user for all the slots in all the domains.

Traditionally, state tracking approaches are based on the assumption that ontology is defined in advance, where all slots and their values are known. Having a predefined ontology can simplify DST into a classification problem and improve performance (Henderson et al., 2014b; Mrkšić et al., 2017; Zhong et al., 2018). However, there are two major drawbacks to this approach: 1) A full ontology is hard to obtain in advance (Xu and Hu, 2018). In the industry, databases are usually exposed through an external API only, which is owned and maintained by others. It is not feasible to gain access to enumerate all the possible values for each slot. 2) Even if a full ontology exists, the number of possible slot values could be large and variable. For example, a restaurant name or a train departure time can contain a large number

*Work partially done while the first author was an intern at Salesforce Research.

of possible values. Therefore, many of the previous works that are based on neural classification models may not be applicable in real scenario.

Budzianowski et al. (2018) recently introduced a multi-domain dialogue dataset (MultiWOZ), which adds new challenges in DST due to its mixed-domain conversations. As shown in Fig. 1, a user can start a conversation by asking to reserve a restaurant, then requests information regarding an attraction nearby, and finally asks to book a taxi. In this case, the DST model has to determine the corresponding domain, slot and value at each turn of dialogue, which contains a large number of combinations in the ontology, i.e., 30 (domain, slot) pairs and over 4,500 possible slot values in total. Another challenge in the multi-domain setting comes from the need to perform multi-turn mapping. Single-turn mapping refers to the scenario where the *(domain, slot, value)* triplet can be inferred from a single turn, while in multi-turn mapping, it should be inferred from multiple turns which happen in different domains. For instance, the *(area, centre)* pair from the *attraction* domain in Fig. 1 can be predicted from the *area* information in the *restaurant* domain, which is mentioned in the preceding turns.

To tackle these challenges, we emphasize that DST models should share tracking knowledge across domains. There are many slots among different domains that share all or some of their values. For example, the *area* slot can exist in many domains, e.g., *restaurant*, *attraction*, and *taxi*. Moreover, the *name* slot in the *restaurant* domain can share the same value with the *departure* slot in the *taxi* domain. Additionally, to enable the DST model to track slots in unseen domains, transferring knowledge across multiple domains is imperative. We expect DST models can learn to track some slots in zero-shot domains by learning to track the same slots in other domains.

In this paper, we propose a transferable dialogue state generator (TRADE) for multi-domain task-oriented dialogue state tracking. The simplicity of our approach and the boost of the performance is the main advantage of TRADE. Contributions in this work are summarized as [1]:

- To overcome the multi-turn mapping problem, TRADE leverages its context-enhanced slot gate and copy mechanism to properly track slot values mentioned anywhere in dialogue history.
- By sharing its parameters across domains, and without requiring a predefined ontology, TRADE can share knowledge between domains to track unseen slot values, achieving state-of-the-art performance on multi-domain DST.
- TRADE enables zero-shot DST by leveraging the domains it has already seen during training. If a few training samples from unseen domains are available, TRADE can adapt to new few-shot domains without forgetting the previous domains.

[1]The code is released at github.com/jasonwu0731/trade-dst

2 TRADE Model

The proposed model in Fig. 2 comprises three components: an utterance encoder, a slot gate, and a state generator. Instead of predicting the probability of every predefined ontology term, our model directly generates slot values. Similar to Johnson et al. (2017) for multilingual neural machine translation, we share all the model parameters, and the state generator starts with a different start-of-sentence token for each *(domain, slot)* pair.

The utterance encoder encodes dialogue utterances into a sequence of fixed-length vectors. To determine whether any of the *(domain, slot)* pairs are mentioned, the context-enhanced slot gate is used with the state generator. The state generator decodes multiple output tokens for all *(domain, slot)* pairs independently to predict their corresponding values. The context-enhanced slot gate predicts whether each of the pairs is actually triggered by the dialogue via a three-way classifier.

Let us define $X = \{(U_1, R_1), \ldots, (U_T, R_T)\}$ as the set of user utterance and system response pairs in T turns of dialogue, and $B = \{B_1, \ldots, B_T\}$ as the dialogue states for each turn. Each B_t is a tuple (domain:D_n, slot:S_m, value:Y_j^{value}), where $D = \{D_1, \ldots, D_N\}$ are the N different domains, and $S = \{S_1, \ldots, S_M\}$ are the M different slots. Assume that there are J possible *(domain, slot)* pairs, and Y_j^{value} is the true word sequence for j-th *(domain ,slot)* pair.

2.1 Utterance Encoder

Note that the utterance encoder can be any existing encoding model. We use bi-directional gated recurrent units (GRU) (Chung et al., 2014) to

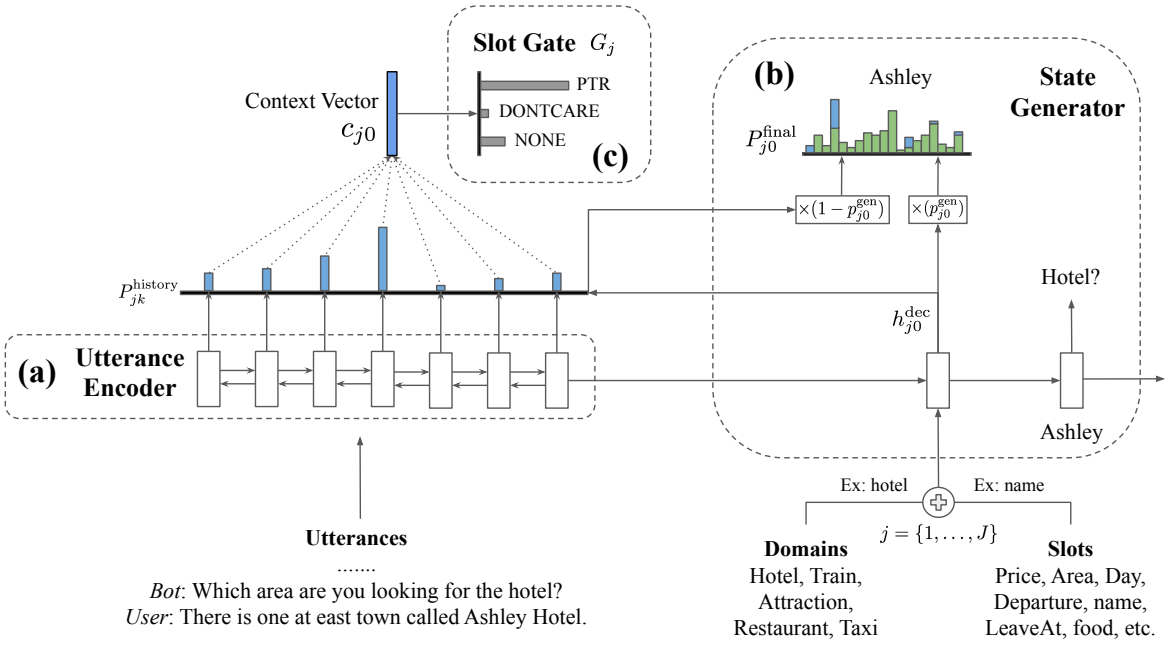


Figure 2: The architecture of the proposed TRADE model, which includes (a) an utterance encoder, (b) a state generator, and (c) a slot gate, all of which are shared among domains. The state generator will decode J times independently for all the possible *(domain, slot)* pairs. At the first decoding step, state generator will take the j-th *(domain, slot)* embeddings as input to generate its corresponding slot values and slot gate. The slot gate predicts whether the j-th *(domain, slot)* pair is triggered by the dialogue.

encode the dialogue history. The input to the utterance encoder is denoted as *history* $X_t = [U_{t-l}, R_{t-l}, \ldots, U_t, R_t] \in \mathbb{R}^{|X_t| \times d_{emb}}$, which is the concatenation of all words in the dialogue history. l is the number of selected dialogue turns and d_{emb} indicates the embedding size. The encoded dialogue history is represented as $H_t = [h^{\text{enc}}_1, \ldots, h^{\text{enc}}_{|X_t|}] \in \mathbb{R}^{|X_t| \times d_{hdd}}$, where d_{hdd} is the hidden size. As mentioned in Section 1, due to the multi-turn mapping problem, the model should infer the states across a sequence of turns. Therefore, we use the recent dialogue history of length l as the utterance encoder input, rather than the current utterance only.

2.2 State Generator

To generate slot values using text from the input source, a copy mechanism is required. There are three common ways to perform copying, i.e., index-based copy (Vinyals et al., 2015), hard-gated copy (Gulcehre et al., 2016; Madotto et al., 2018; Wu et al., 2019) and soft-gated copy (See et al., 2017; McCann et al., 2018). The index-based mechanism is not suitable for DST task because the exact word(s) of the true slot value are not always found in the utterance. The hard-gate copy mechanism usually needs additional supervision on the gating function. As such, we employ soft-gated pointer-generator copying to combine a distribution over the vocabulary and a distribution over the dialogue history into a single output distribution.

We use a GRU as the decoder of the state generator to predict the value for each *(domain, slot)* pair, as shown in Fig. 2. The state generator decodes J pairs independently. We simply supply the summed embedding of the domain and slot as the first input to the decoder. At decoding step k for the j-th *(domain, slot)* pair, the generator GRU takes a word embedding w_{jk} as its input and returns a hidden state h^{dec}_{jk}. The state generator first maps the hidden state h^{dec}_{jk} into the vocabulary space P^{vocab}_{jk} using the trainable embedding $E \in \mathbb{R}^{|V| \times d_{hdd}}$, where $|V|$ is the vocabulary size. At the same time, the h^{dec}_{jk} is used to compute the history attention P^{history}_{jk} over the encoded dialogue history H_t:

$$\begin{aligned} P^{\text{vocab}}_{jk} &= \text{Softmax}(E \cdot (h^{\text{dec}}_{jk})^\top) \in \mathbb{R}^{|V|}, \\ P^{\text{history}}_{jk} &= \text{Softmax}(H_t \cdot (h^{\text{dec}}_{jk})^\top) \in \mathbb{R}^{|X_t|}. \end{aligned} \quad (1)$$

The final output distribution P^{final}_{jk} is the weighted-

sum of two distributions,

$$P_{jk}^{\text{final}} = p_{jk}^{\text{gen}} \times P_{jk}^{\text{vocab}} + (1 - p_{jk}^{\text{gen}}) \times P_{jk}^{\text{history}} \in \mathbb{R}^{|V|}. \quad (2)$$

The scalar p_{jk}^{gen} is trainable to combine the two distributions, which is computed by

$$p_{jk}^{\text{gen}} = \text{Sigmoid}(W_1 \cdot [h_{jk}^{\text{dec}}; w_{jk}; c_{jk}]) \in \mathbb{R}^1, \quad c_{jk} = P_{jk}^{\text{history}} \cdot H_t \in \mathbb{R}^{d_{hdd}} \quad (3)$$

where W_1 is a trainable matrix and c_{jk} is the context vector. Note that due to Eq (2), our model is able to generate words even if they are not predefined in the vocabulary.

2.3 Slot Gate

Unlike single-domain DST problems, where only a few slots that need to be tracked, e.g., four slots in WOZ (Wen et al., 2017), and eight slots in DSTC2 (Henderson et al., 2014a), there are a large number of *(domain, slot)* pairs in multi-domain DST problems. Therefore, the ability to predict the domain and slot at current turn t becomes more challenging.

Our context-enhanced slot gate G is a simple three-way classifier that maps a context vector taken from the encoder hidden states H_t to a probability distribution over *ptr*, *none*, and *dontcare* classes. For each *(domain, slot)* pair, if the slot gate predicts *none* or *dontcare*, we ignore the values generated by the decoder and fill the pair as "not-mentioned" or "does not care". Otherwise, we take the generated words from our state generator as its value. With a linear layer parameterized by $W_g \in \mathbb{R}^{3 \times d_{hdd}}$, the slot gate for the j-th *(domain, slot)* pair is defined as

$$G_j = \text{Softmax}(W_g \cdot (c_{j0})^\top) \in \mathbb{R}^3, \quad (4)$$

where c_{j0} is the context vector computed in Eq (3) using the first decoder hidden state.

2.4 Optimization

During training, we optimize for both the slot gate and the state generator. For the former, the cross-entropy loss L_g is computed between the predicted slot gate G_j and the true one-hot label y_j^{gate},

$$L_g = \sum_{j=1}^{J} -\log(G_j \cdot (y_j^{\text{gate}})^\top). \quad (5)$$

For the latter, another cross-entropy loss L_v between P_{jk}^{final} and the true words Y_j^{label} is used. We define L_v as

$$L_v = \sum_{j=1}^{J} \sum_{k=1}^{|Y_j|} -\log(P_{jk}^{\text{final}} \cdot (y_{jk}^{\text{value}})^\top). \quad (6)$$

L_v is the sum of losses from all the *(domain, slot)* pairs and their decoding time steps. We optimize the weighted-sum of these two loss functions using hyper-parameters α and β,

$$L = \alpha L_g + \beta L_v. \quad (7)$$

3 Unseen Domain DST

In this section, we focus on the ability of TRADE to generalize to an unseen domain by considering zero-shot transferring and few-shot domain expanding. In the zero-shot setting, we assume we have no training data in the new domain, while in the few-shot case, we assume just 1% of the original training data in the unseen domain is available (around 20 to 30 dialogues). One of the motivations to perform unseen domain DST is because collecting a large-scale task-oriented dataset for a new domain is expensive and time-consuming (Budzianowski et al., 2018), and there are a large amount of domains in realistic scenarios.

3.1 Zero-shot DST

Ideally, based on the slots already learned, a DST model is able to directly track those slots that are present in a new domain. For example, if the model is able to track the *departure* slot in the *train* domain, then that ability may transfer to the *taxi* domain, which uses similar slots. Note that generative DST models take the dialogue context/history X, the domain D, and the slot S as input and then generate the corresponding values Y^{value}. Let $(X, D_{\text{source}}, S_{\text{source}}, Y_{\text{source}}^{\text{value}})$ be the set of samples seen during the training phase and $(X, D_{\text{target}}, S_{\text{target}}, Y_{\text{target}}^{\text{value}})$ the samples which the model was not trained to track. A zero-shot DST model should be able to generate the correct values of $Y_{\text{target}}^{\text{value}}$ given the context X, domain D_{target}, and slot S_{target}, without using any training samples. The same context X may appear in both source and target domains but the pairs $(D_{\text{target}}, S_{\text{target}})$ are unseen. This setting is extremely challenging if no slot in S_{target} appears in S_{source}, since the model has never been trained to track such a slot.

3.2 Expanding DST for Few-shot Domain

In this section, we assume that only a small number of samples from the new domain $(X, D_{\text{target}}, S_{\text{target}}, Y^{\text{value}}_{\text{target}})$ are available, and the purpose is to evaluate the ability of our DST model to transfer its learned knowledge to the new domain without forgetting previously learned domains. There are two advantages to performing few-shot domain expansion: 1) being able to quickly adapt to new domains and obtain decent performance with only a small amount of training data; 2) not requiring retraining with all the data from previously learned domains, since the data may no longer be available and retraining is often very time-consuming.

Firstly, we consider a straightforward naive baseline, i.e., fine-tuning with no constraints. Then, we employ two specific continual learning techniques: elastic weight consolidation (EWC) (Kirkpatrick et al., 2017) and gradient episodic memory (GEM) (Lopez-Paz et al., 2017) to fine-tune our model. We define Θ_S as the model's parameters trained in the source domain, and Θ indicates the current optimized parameters according to the target domain data.

EWC uses the diagonal of the Fisher information matrix F as a regularizer for adapting to the target domain data. This matrix is approximated using samples from the source domain. The EWC loss is defined as

$$L_{ewc}(\Theta) = L(\Theta) + \sum_i \frac{\lambda}{2} F_i(\Theta_i - \Theta_{S,i})^2, \quad (8)$$

where λ is a hyper-parameter. Different from EWC, GEM keeps a small number of samples K from the source domains, and, while the model learns the new target domain, a constraint is applied on the gradient to prevent the loss on the stored samples from increasing. The training process is defined as:

$$\begin{aligned} &\text{Minimize}_\Theta \; L(\Theta) \\ &\text{Subject to } L(\Theta, K) \leq L(\Theta_S, K), \end{aligned} \quad (9)$$

where $L(\Theta, K)$ is the loss value of the K stored samples. Lopez-Paz et al. (2017) show how to solve the optimization problem in Eq (9) with quadratic programming if the loss of the stored samples increases.

	Hotel	Train	Attraction	Restaurant	Taxi
Slots	price, type, parking, stay, day, people, area, stars, internet, name	destination, departure, day, arrive by, leave at, people	area, name, type	food, price, area, name, time, day, people	destination, departure, arrive by, leave by
Train	3381	3103	2717	3813	1654
Valid	416	484	401	438	207
Test	394	494	395	437	195

Table 1: The dataset information of MultiWOZ. In total, there are 30 *(domain, slot)* pairs from the selected five domains. The numbers in the last three rows indicate the number of dialogues for train, validation and test sets.

4 Experiments

4.1 Dataset

Multi-domain Wizard-of-Oz (Budzianowski et al., 2018) (MultiWOZ) is the largest existing human-human conversational corpus spanning over seven domains, containing 8438 multi-turn dialogues, with each dialogue averaging 13.68 turns. Different from existing standard datasets like WOZ (Wen et al., 2017) and DSTC2 (Henderson et al., 2014a), which contain less than 10 slots and only a few hundred values, MultiWOZ has 30 *(domain, slot)* pairs and over 4,500 possible values. We use the DST labels from the original training, validation and testing dataset. Only five domains (*restaurant*, *hotel*, *attraction*, *taxi*, *train*) are used in our experiment because the other two domains (*hospital*, *police*) have very few dialogues (10% compared to others) and only appear in the training set. The slots in each domain and the corresponding data size are reported in Table 1.

4.2 Training Details

Multi-domain Joint Training The model is trained end-to-end using the Adam optimizer (Kingma and Ba, 2015) with a batch size of 32. The learning rate annealing is in the range of $[0.001, 0.0001]$ with a dropout ratio of 0.2. Both α and β in Eq (7) are set to one. All the embeddings are initialized by concatenating Glove embeddings (Pennington et al., 2014) and character embeddings (Hashimoto et al., 2016), where the dimension is 400 for each vocabulary word. A greedy search decoding strategy is used for our state generator since the generated slot values are usually short in length. In addition, to in-

crease model generalization and simulate an out-of-vocabulary setting, a word dropout is utilized with the utterance encoder by randomly masking a small amount of input tokens, similar to Bowman et al. (2016).

Domain Expanding For training, we follow the same procedure as in the joint training section, and we run a small grid search for all the methods using the validation set. For EWC, we set different values of λ for all the domains, and the optimal value is selected using the validation set. Finally, in GEM, we set the memory sizes K to 1% of the source domains.

4.3 Results

Two evaluation metrics, joint goal accuracy and slot accuracy, are used to evaluate the performance on multi-domain DST. The joint goal accuracy compares the predicted dialogue states to the ground truth B_t at each dialogue turn t, and the output is considered correct if and only if all the predicted values exactly match the ground truth values in B_t. The slot accuracy, on the other hand, individually compares each (domain, slot, value) triplet to its ground truth label.

Multi-domain Training We make a comparison with the following existing models: MDBT (Ramadan et al., 2018), GLAD (Zhong et al., 2018), GCE (Nouri and Hosseini-Asl, 2018), and SpanPtr (Xu and Hu, 2018), and we briefly describe these baselines models below:

- MDBT [2]: Multiple bi-LSTMs are used to encode system and user utterances. The semantic similarity between utterances and every predefined ontology term is computed separately. Each ontology term is triggered if the predicted score is greater than a threshold.

- GLAD [3]: This model uses self-attentive RNNs to learn a global tracker that shares parameters among slots and a local tracker that tracks each slot. The model takes previous system actions and the current user utterance as input, and computes semantic similarity with predefined ontology terms.

- GCE: This is the current state-of-the-art model on the single-domain WOZ dataset (Wen et al., 2017). It is a simplified and speed up version of GLAD without slot-specific RNNs.

- SpanPtr: Most related to our work, this is the first model that applies pointer networks (Vinyals et al., 2015) to the single-domain DST problem, which generates both start and end pointers to perform index-based copying.

[2] github.com/osmanio2/multi-domain-belief-tracking

[3] github.com/salesforce/glad

	MultiWOZ		**MultiWOZ (Only Restaurant)**	
	Joint	***Slot***	***Joint***	***Slot***
MDBT	15.57	89.53	17.98	54.99
GLAD	35.57	95.44	53.23	96.54
GCE	36.27	98.42	60.93	95.85
SpanPtr	30.28	93.85	49.12	87.89
TRADE	**48.62**	96.92	**65.35**	93.28

Table 2: The multi-domain DST evaluation on MultiWOZ and its single *restaurant* domain. TRADE has the highest joint accuracy, which surpasses current state-of-the-art GCE model.

To have a fair comparison, we modify the original implementation of the MDBT and GLAD models by: 1) adding *name*, *destination*, and *departure* slots for evaluation if they were discarded or replaced by placeholders; and 2) removing the hand-crafted rules of tracking the booking slots such as *stay* and *people* slots if there are any; and 3) creating a full ontology for their model to cover all *(domain, slot, value)* pairs that were not in the original ontology generated by the data provider.

As shown in Table 2, TRADE achieves the highest performance, 48.62% on joint goal accuracy and 96.92% on slot accuracy, on MultiWOZ. For comparison with the performance on single-domain, the results on the *restaurant* domain of MultiWOZ are reported as well. The performance difference between SpanPtr and our model mainly comes from the limitation of index-based copying. For examples, if the true label for the price range slot is *cheap*, the relevant user utterance describing the restaurant may actually be, for example, *economical*, *inexpensive*, or *cheaply*. Note that the MDBT, GLAD, and GCE models each need a predefined domain ontology to perform binary classification for each ontology term, which hinders their DST tracking performance, as mentioned in Section 1.

We visualize the cosine similarity matrix for all possible slot embeddings in Fig. 3. Most of the

Evaluation on 4 Domains		Joint *Except Hotel*	Slot *Except Hotel*	Joint *Except Train*	Slot *Except Train*	Joint *Except Attraction*	Slot *Except Attraction*	Joint *Except Restaurant*	Slot *Except Restaurant*	Joint *Except Taxi*	Slot *Except Taxi*
Base Model (BM) training on 4 domains		58.98	96.75	55.26	96.76	55.02	97.03	54.69	96.64	49.87	96.77
Fine-tuning BM on 1% new domain	*Naive*	36.08	93.48	23.25	90.32	40.05	95.54	32.85	91.69	46.10	96.34
	EWC	40.82	94.16	28.02	91.49	45.37	84.94	34.45	92.53	**46.88**	96.44
	GEM	**53.54**	**96.27**	**50.69**	**96.42**	**50.51**	**96.66**	**45.91**	**95.58**	46.43	**96.45**
Evaluation on New Domain		*Hotel*		*Train*		*Attraction*		*Restaurant*		*Taxi*	
Training 1% New Domain		19.53	77.33	44.24	85.66	**35.88**	**68.60**	32.72	82.39	60.38	72.82
Fine-tuning BM on 1% new domain	*Naive*	19.13	75.22	**59.83**	**90.63**	29.39	60.73	**42.42**	**86.82**	**63.81**	**79.81**
	EWC	19.35	76.25	58.10	90.33	32.28	62.43	40.93	85.80	63.61	79.65
	GEM	**19.73**	**77.92**	54.31	89.55	34.73	64.37	39.24	86.05	63.16	79.27

Table 3: We run domain expansion experiments by excluding one domain and fine-tuning on that domain. The first row is the base model trained on the four domains. The second row is the results on the four domains after fine-tuning on 1% new domain data using three different strategies. One can find out that GEM outperforms Naive and EWC fine-tuning in terms of catastrophic forgetting on the four domains. Then, we evaluate the results on new domain for two cases: training from scratch and fine-tuning from the base model. Results show that fine-tuning from the base model usually achieves better results on the new domain compared to training from scratch.

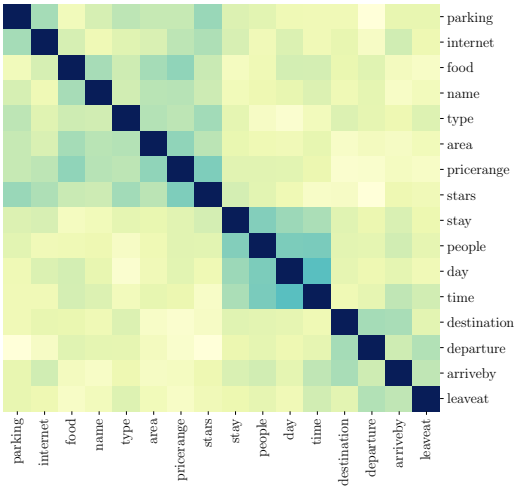


Figure 3: Embeddings cosine similarity visualization. The rows and columns are all the possible slots in MultiWOZ. Slots that share similar values or have correlated values learn similar embeddings. For example *destination vs. departure* (which share similar values) or *price range vs. stars* exhibit high correlation.

slot embeddings are not close to each other, which is expected because the model only depends on these features as start-of-sentence embeddings to distinguish different slots. Note that some slots are relatively close because either the values they track may share similar semantic meanings or the slots are correlated. For example, *destination* and *departure* track names of cities, while *people* and *stay* track numbers. On the other hand, *price range* and *star* in hotel domain are correlated because high-star hotels are usually expensive.

	Trained Single		Zero-Shot	
	Joint	*Slot*	*Joint*	*Slot*
Hotel	55.52	92.66	13.70	65.32
Train	77.71	95.30	22.37	49.31
Attraction	71.64	88.97	19.87	55.53
Restaurant	65.35	93.28	11.52	53.43
Taxi	76.13	89.53	**60.58**	73.92

Table 4: Zero-shot experiments on an unseen domain. In *taxi* domain, our model achieves 60.58% joint goal accuracy without training on any samples from *taxi* domain. *Trained Single* column is the results achieved by training on 100% single-domain data as a reference.

Zero-shot We run zero-shot experiments by excluding one domain from the training set. As shown in Table 4, the *taxi* domain achieves the highest zero-shot performance, 60.58% on joint goal accuracy, which is close to the result achieved by training on all the *taxi* domain data (76.13%). Although performances on the other zero-shot domains are not especially promising, they still achieve around 50 to 65% slot accuracy without using any in-domain samples. The reason why the zero-shot performance on the *taxi* domain is high is because all four slots share similar values with the corresponding slots in the *train* domain.

Domain Expanding In this setting, the TRADE model is pre-trained on four domains and a *held-out* domain is reserved for domain expansion to perform fine-tuning. After fine-tuning on the new domain, we evaluate the performance of TRADE on 1) the four pre-trained domains and 2) the new domain. We experiment with different fine-tuning

strategies. The *base model* row in Table 3 indicates the results evaluated on the four domains using their in-domain training data, and the *Training 1% New Domain* row indicates the results achieved by training from scratch using 1% of the new domain data. In general, GEM outperforms naive and EWC fine-tuning in terms of overcoming catastrophic forgetting. We also find that pre-training followed by fine-tuning outperforms training from scratch on the single domain.

Fine-tuning TRADE with GEM maintains higher performance on the original four domains. Take the *hotel* domain as an example, the performance on the four domains after fine-tuning with GEM only drops from 58.98% to 53.54% (-5.44%) on joint accuracy, whereas naive fine-tuning deteriorates the tracking ability, dropping joint goal accuracy to 36.08% (-22.9%).

Expanding TRADE from four domains to a new domain achieves better performance than training from scratch on the new domain. This observation underscores the advantages of transfer learning with the proposed TRADE model. For example, our TRADE model achieves 59.83% joint accuracy after fine-tuning using only 1% of *Train* domain data, outperforming the training *Train* domain from scratch, which achieves 44.24% using the same amount of new-domain data.

Finally, when considering *hotel* and *attraction* as new domain, fine-tuning with GEM outperforms the naive fine-tuning approach on the new domain. To elaborate, GEM obtains 34.73% joint accuracy on the *attraction* domain, but naive fine-tuning on that domain can only achieve 29.39%. This implies that in some cases learning to keep the tracking ability (learned parameters) of the learned domains helps to achieve better performance for the new domain.

5 Error Analysis

An error analysis of multi-domain training is shown in Fig. 4. Not surprisingly, *name* slots in the *restaurant*, *attraction*, and *hotel* domains have the highest error rates, 8.50%, 8.17%, and 7.86%, respectively. It is because this slot usually has a large number of possible values that is hard to recognize. On the other hand, number-related slots such as *arrive_by*, *people*, and *stay* usually have the lowest error rates. We also find that the *type* slot of *hotel* domain has a high error rate, even if it is an easy task with only two possible values in

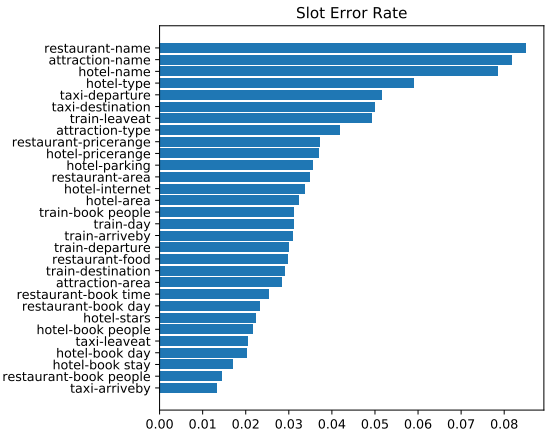
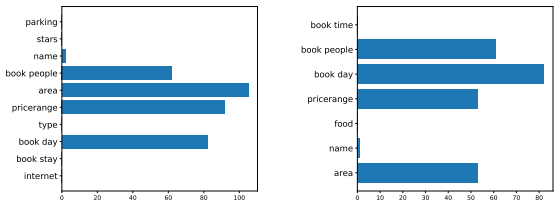


Figure 4: Slots error rate on test set of multi-domain training. The *name* slot in *restaurant* domain has the highest error rate, 8.50%, and the *arrive_by* slot in *taxi* domain has the lowest error rate, 1.33%



(a) Hotel (b) Restaurant

Figure 5: Zero-shot DST error analysis on (a) *hotel* and (b) restaurant domains. The x-axis represents the correctness number of each slot which has non-empty values. In *hotel* domain, abilities of tracking *people, area, price_range*, and *day* slots are successfully transferred.

the ontology. The reason is that labels of the *(hotel, type)* pair are usually missing in the dataset, which makes our prediction incorrect even if it is supposed to be predicted.

In Fig. 5, we show the zero-shot analysis of two selected domains, *hotel* and *restaurant*, that have more slots to be tracked. To better understand the behavior of knowledge transferring, here we only take labels that are not missing into account, i.e., we ignore data that is labeled as "none" because predicting "none" is relatively easier for the model. In both *hotel* and *restaurant* domains, *people, area, price_range*, and *day* slots are successfully transferred from the other four domains. For unseen slots that only appear in one domain, it is very hard for our model to track correctly. For example, *parking, stars* and *internet* slots are only appeared in *hotel* domain, and the *food* slot is unique to the *restaurant* domain.

6 Related Work

Dialogue State Tracking Traditional dialogue state tracking models combine semantics extracted by language understanding modules to estimate the current dialogue states (Williams and Young, 2007; Thomson and Young, 2010; Wang and Lemon, 2013; Williams, 2014), or to jointly learn speech understanding (Henderson et al., 2014b; Zilka and Jurcicek, 2015; Wen et al., 2017). One drawback is that they rely on hand-crafted features and complex domain-specific lexicons (besides the ontology), and are difficult to extend and scale to new domains.

Mrkšić et al. (2017) use distributional representation learning to leverage semantic information from word embeddings to and resolve lexical/morphological ambiguity. However, parameters are not shared across slots. On the other hand, Nouri and Hosseini-Asl (2018) utilizes global modules to share parameters between slots, and Zhong et al. (2018) uses slot-specific local modules to learn slot features, which has proved to successfully improve tracking of rare slot values. Lei et al. (2018) use a Seq2Seq model to generate belief spans and the delexicalized response at the same time. Ren et al. (2018) propose StateNet that generates a dialogue history representation and compares the distances between this representation and value vectors in the candidate set. Xu and Hu (2018) use the index-based pointer network for different slots, and show the ability to point to unknown values. However, many of them require a predefined domain ontology, and the models were only evaluated on single-domain setting (DSTC2).

For multi-domain DST, Rastogi et al. (2017) propose a multi-domain approach using two-layer bi-GRU. Although it does not need an ad-hoc state update mechanism, it relies on delexicalization to extract the features. Ramadan et al. (2018) propose a model to jointly track domain and the dialogue states using multiple bi-LSTM. They utilize semantic similarity between utterances and the ontology terms and allow the information to be shared across domains. For a more general overview, readers may refer to the neural dialogue review paper from Gao et al. (2018).

Zero/Few-Shot and Continual Learning Different components of dialogue systems have previously been used for zero-shot application, e.g., intention classifiers (Chen et al., 2016), slot-filling (Bapna et al., 2017), and dialogue policy (Gašić and Young, 2014). For language generation, Johnson et al. (2017) propose single encoder-decoder models for zero-shot machine translation, and Zhao and Eskenazi (2018) propose cross-domain zero-shot dialogue generation using action matching. Moreover, few-shot learning in natural language applications has been applied in semantic parsing (Huang et al., 2018), machine translation (Gu et al., 2018), and text classification (Yu et al., 2018) with meta-learning approaches (Schmidhuber, 1987; Finn et al., 2017). These tasks usually have multiple tasks to perform fast adaptation, instead in our case the number of existing domains are limited. Lastly, several approaches have been proposed for continual learning in the machine learning community (Kirkpatrick et al., 2017; Lopez-Paz et al., 2017; Rusu et al., 2016; Fernando et al., 2017; Lee et al., 2017), especially in image recognition tasks (Aljundi et al., 2017; Rannen et al., 2017). The applications within NLP has been comparatively limited, e.g., Shu et al. (2016, 2017b) for opinion mining, Shu et al. (2017a) for document classification, and Lee (2017) for hybrid code networks (Williams et al., 2017).

7 Conclusion

We introduce a transferable dialogue state generator for multi-domain dialogue state tracking, which learns to track states without any predefined domain ontology. TRADE shares all of its parameters across multiple domains and achieves state-of-the-art joint goal accuracy and slot accuracy on the MultiWOZ dataset for five different domains. Moreover, domain sharing enables TRADE to perform zero-shot DST for unseen domains and to quickly adapt to few-shot domains without forgetting the learned ones. In future work, transferring knowledge from other resources can be applied to further improve zero-shot performance, and collecting a dataset with a large number of domains is able to facilitate the application and study of meta-learning techniques within multi-domain DST.

Acknowledgments

This work is partially funded by MRP/055/18 of the Innovation Technology Commission, of the Hong Kong University of Science and Technology (HKUST).

References

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. 2017. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3375.

Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2017. Towards zero-shot frame semantic parsing for domain scaling. *arXiv preprint arXiv:1707.02363*.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6045–6049. IEEE.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. 2017. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374. ACM.

Milica Gašić and Steve Young. 2014. Gaussian processes for pomdp-based dialogue manager optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631. Association for Computational Linguistics.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. *arXiv preprint arXiv:1603.08148*.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014b. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299.

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. 2018. Natural language to structured query generation via meta-learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 732–738. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, page 201611835.

Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*, pages 4652–4662.

Sungjin Lee. 2017. Toward continual learning for conversational agents. *arXiv preprint arXiv:1712.09943*.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1437–1447.

David Lopez-Paz et al. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1468–1478.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking model. In *Advances in neural information processing systems (NeurIPS), 2nd Conversational AI workshop*. https://arxiv.org/abs/1812.00899.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437. Association for Computational Linguistics.

Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. 2017. Encoder based lifelong learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1320–1328.

Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 561–568. IEEE.

Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. 2018. Towards universal dialogue state tracking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2780–2786.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Jurgen Schmidhuber. 1987. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Technische Universitat Munchen, Germany, 14 May.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1073–1083.

Lei Shu, Bing Liu, Hu Xu, and Annice Kim. 2016. Lifelong-rl: Lifelong relaxation labeling for separating entities and aspects in opinion targets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 225. NIH Public Access.

Lei Shu, Hu Xu, and Bing Liu. 2017a. Doc: Deep open classification of text documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2911–2916. Association for Computational Linguistics.

Lei Shu, Hu Xu, and Bing Liu. 2017b. Lifelong learning crf for supervised aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 148–154. Association for Computational Linguistics.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M. Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of*

the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 438–449. Association for Computational Linguistics.

Jason D Williams. 2014. Web-style ranking and slu combination for dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 282–291.

Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 665–677. Association for Computational Linguistics.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019. Global-to-local memory pointer networks for task-oriented dialogue. In *Proceedings of the 7th International Conference on Learning Representations*.

Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1457. Association for Computational Linguistics.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. 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 1206–1215. Association for Computational Linguistics.

Tiancheng Zhao and Maxine Eskenazi. 2018. Zero-shot dialog generation with cross-domain latent actions. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 1–10.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467. Association for Computational Linguistics.

Lukas Zilka and Filip Jurcicek. 2015. Incremental lstm-based dialog state tracker. In *2015 Ieee Workshop on Automatic Speech Recognition and Understanding (Asru)*, pages 757–762. IEEE.

Emotion-Cause Pair Extraction: A New Task to Emotion Analysis in Texts

Rui Xia, Zixiang Ding
School of Computer Science and Engineering,
Nanjing University of Science and Technology, China
{rxia, dingzixiang}@njust.edu.cn

Abstract

Emotion cause extraction (ECE), the task aimed at extracting the potential causes behind certain emotions in text, has gained much attention in recent years due to its wide applications. However, it suffers from two shortcomings: 1) the emotion must be annotated before cause extraction in ECE, which greatly limits its applications in real-world scenarios; 2) the way to first annotate emotion and then extract the cause ignores the fact that they are mutually indicative. In this work, we propose a new task: emotion-cause pair extraction (ECPE), which aims to extract the potential pairs of emotions and corresponding causes in a document. We propose a 2-step approach to address this new ECPE task, which first performs individual emotion extraction and cause extraction via multi-task learning, and then conduct emotion-cause pairing and filtering. The experimental results on a benchmark emotion cause corpus prove the feasibility of the ECPE task as well as the effectiveness of our approach.

1 Introduction

Emotion cause extraction (ECE) aims at extracting potential causes that lead to emotion expressions in text. The ECE task was first proposed and defined as a word-level sequence labeling problem in Lee et al. (2010). To solve the shortcoming of extracting causes at word level, Gui et al. (2016a) released a new corpus which has received much attention in the following study and become a benchmark dataset for ECE research.

Figure 1 displays an example from this corpus, There are five clauses in a document. The emotion "happy" is contained in the fourth clause. We denote this clause as *emotion clause*, which refers to a clause that contains emotions. It has two corresponding causes: "a policeman visited the old man with the lost money" in the second clause, and "told him that the thief was caught" in the third clause. We denote them as *cause clause*, which refers to a clause that contains causes.

The ECE task was formalized as a clause-level binary classification problem in Gui et al. (2016a). The goal is to detect for each clause in a document, whether this clause is a cause given the annotation of emotion. This framework was followed by most of the recent studies in this field (Lee et al., 2010; Gui et al., 2016a; Li et al., 2018; Xu et al., 2019; Yu et al., 2019).

However, there are two shortcomings in the current ECE task. The first is that emotions must be annotated before cause extraction in the test set, which limits the applications of ECE in real-world scenarios. The second is that the way to first annotate the emotion and then extract the cause ignores the fact that emotions and causes are mutually indicative.

In this work, we propose a new task: emotion-cause pair extraction (ECPE), which aims to extract all potential pairs of emotions and corresponding causes in a document. In Figure 1 we show the difference between the traditional ECE task and our new ECPE task. The goal of ECE is to extract the corresponding cause clause of the given emotion. In addition to a document as the input, ECE needs to provide annotated emotion at first before cause extraction. In contrast, the output of our ECPE task is a pair of emotion-cause, without the need of providing emotion annotation in advance. Take Figure 1 for example, given the annotation of emotion: "happy", the goal of ECE is to track the two corresponding cause clauses: "a policeman visited the old man with the lost money" and "and told him that the thief was caught". While in the ECPE task, the goal is to directly extract all pairs of emotion clause and cause clause, including ("The old man was very happy", "a policeman visited the old man with the lost money") and ("The old man was very happy", "and told him that the thief was caught"), without

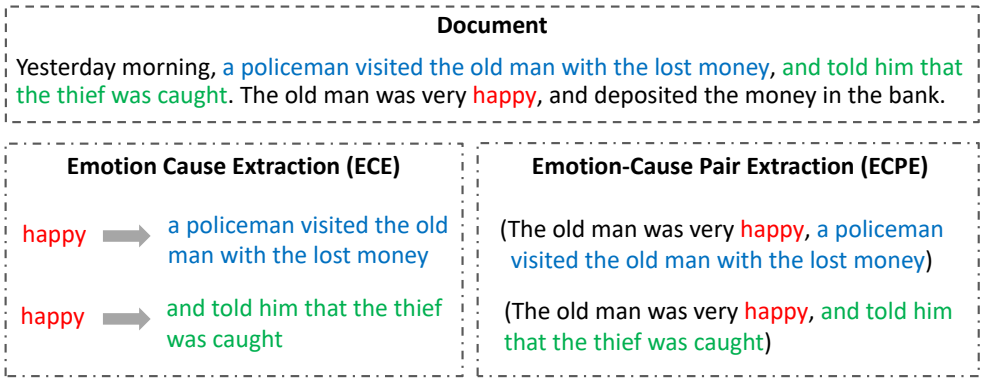


Figure 1: An example showing the difference between the ECE task and the ECPE task.

providing the emotion annotation "happy".

To address this new ECPE task, we propose a two-step framework. Step 1 converts the emotion-cause pair extraction task to two individual subtasks (emotion extraction and cause extraction respectively) via two kinds of multi-task learning networks, with the goal to extract a set of emotion clauses and a set of cause clauses. Step 2 performs emotion-cause pairing and filtering. We combine all the elements of the two sets into pairs and finally train a filter to eliminate the pairs that do not contain a causal relationship.

We evaluated our approach based on a benchmark emotion cause dataset (Gui et al., 2016a) without using emotion annotations on the test data. We finally achieve the F1 score of 61.28% in emotion-cause pair extraction. The experimental results prove the feasibility of the ECPE task and the effectiveness of our approach.

In addition to the emotion-cause pair extraction evaluation, we also evaluate the performance on two individual tasks (emotion extraction and cause extraction). Without relying on the emotion annotations on the test set, our approach achieves comparable cause extraction performance to traditional ECE methods (slightly lower than the state-of-the-art). In comparison with the traditional ECE methods that removes the emotion annotation dependence, our approach shows great advantages.

The main contributions of this work can be summarized as follows:

- We propose a new task: emotion-cause pair extraction (ECPE). It solves the shortcomings of the traditional ECE task that depends on the annotation of emotion before extracting cause, and allows emotion cause analysis to be applied to real-world scenarios.
- We propose a two-step framework to address the ECPE task, which first performs individual emotion extraction and cause extraction and then conduct emotion-cause pairing and filtering.
- Based on a benchmark ECE corpus, we construct a corpus suitable for the ECPE task. The experimental results prove the feasibility of the ECPE task as well as the effectiveness of our approach.

2 Related Work

Lee et al. (2010) first presented the task of emotion cause extraction (ECE) and defined this task as extracting the word-level causes that lead to the given emotions in text. They constructed a small-scale Chinese emotion cause corpus in which the spans of both emotion and cause were annotated. Based on the same task settings, there were some other individual studies that conducted ECE research on their own corpus using rule based methods (Neviarouskaya and Aono, 2013; Li and Xu, 2014; Gao et al., 2015a,b; Yada et al., 2017) or machine learning methods (Ghazi et al., 2015; Song and Meng, 2015).

Chen et al. (2010) suggested that a clause may be the most appropriate unit to detect causes based on the analysis of the corpus in (Lee et al., 2010), and transformed the task from word-level to clause-level. They proposed a multi-label approach that detects multi-clause causes and captures the long-distance information. There were a lot of work based on this task setting. Russo et al. (2011) introduced a method based on the linguistic patterns and common sense knowledge for the identification of Italian sentences which contain a

cause phrase. Gui et al. (2014) used 25 manually complied rules as features, and chose machine learning models, such as SVM and CRFs, to detect causes. Gui et al. (2016a), Gui et al. (2016b) and Xu et al. (2017) released a Chinese emotion cause dataset using SINA city news. This corpus has received much attention in the following study and has become a benchmark dataset for ECE research. Based on this corpus, several traditional machine learning methods (Gui et al., 2016a,b; Xu et al., 2017) and deep learning methods (Gui et al., 2017; Li et al., 2018; Yu et al., 2019; Xu et al., 2019) were proposed.

In addition, Cheng et al. (2017) focused on cause detection for Chinese microblogs using a multiple-user structure. They formalized two cause detection tasks for microblogs (current-subtweet-based cause detection and original-subtweet-based cause detection) and introduced SVM and LSTM to deal with them. Chen et al. (2018b) presented a neural network-based joint approach for emotion classification and cause detection in order to capture mutual benefits across these two sub-tasks. Chen et al. (2018a) proposed a hierarchical Convolution Neural Network (Hier-CNN), which used clause-level encoder and subtweet-level encoder to incorporate the word context features and event-based features respectively.

All of the above work attempts to extract word-level or clause-level causes given the emotion annotations. While our work is different from them, we propose to extract both the emotion and the corresponding causes at the same time (i.e., emotion-cause pair extraction) and to investigate whether indicating causes can improve emotion extraction and vice versa. Since we believe that cause and emotion are not mutually independent.

3 Task

First of all, we give the definition of our emotion-cause pair extraction (ECPE) task. Given a document consisting of multiple clauses $d = [c_1, c_2, ..., c_{|d|}]$, the goal of ECPE is to extract a set of emotion-cause pairs in d:

$$P = \{\cdots, (c^e, c^c), \cdots\}, \quad (1)$$

where c^e is an emotion clause and c^c is the corresponding cause clause

In traditional emotion cause extraction task, the goal is to extract c^c given the annotation of c^e : $c^e \rightarrow c^c$. In comparison, the ECPE task is new and more difficult to address, because the annotation of emotion c^e is not provided before extraction.

Note that similar as the traditional ECE task, the ECPE task is also defined at the clause level, due to the difficulty describing emotion causes at the word/phrase level. It means that the "emotion" and "cause" used in this paper refer to "emotion clause" and "cause clause" respectively.

4 Approach

In this work, we propose a two-step approach to address this new ECPE task:

- **Step 1 (Individual Emotion and Cause Extraction)**. We first convert the emotion-cause pair extraction task to two individual sub-tasks (emotion extraction and cause extraction respectively). Two kinds of multi-task learning networks are proposed to model the two sub-tasks in a unified framework, with the goal to extract a set of emotion clauses $E = \{c^e_1, \cdots, c^e_m\}$ and a set of cause clauses $C = \{c^c_1, \cdots, c^c_n\}$ for each document.
- **Step 2 (Emotion-Cause Pairing and Filtering)**. We then pair the emotion set E and the cause set C by applying a Cartesian product to them. This yields a set of candidate emotion-cause pairs. We finally train a filter to eliminate the pairs that do not contain a causal relationship between emotion and cause.

4.1 Step 1: Individual Emotion and Cause Extraction

The goal of Step 1 is to extract a set of emotion clauses and a set of cause clauses for each document, respectively. To this end, we propose two kinds of multi-task learning networks, (i.e., Independent Multi-task Learning and Interactive Multi-task Learning). The latter is an enhanced version that further captures the correlation between emotion and cause on the basis of the former.

4.1.1 Independent Multi-task Learning

In our task, a document contains multiple clauses: $d = [c_1, c_2, ..., c_{|d|})]$, and each c_i also contains multiple words $c_i = [w_{i,1}, w_{i,2}, ..., w_{i,|c_i|}]$. To capture such a "word-clause-document" structure, we employ a Hierarchical Bi-LSTM network which contains two layers, as shown in Figure 2.

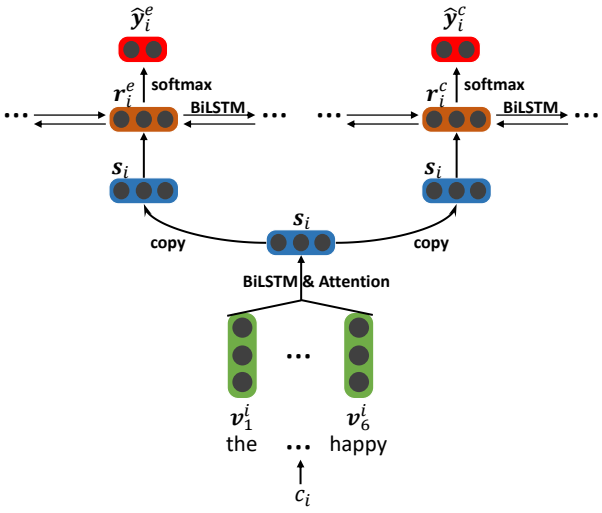


Figure 2: The Model for Independent Multi-task Learning (Indep).

The lower layer consists of a set of word-level Bi-LSTM modules, each of which corresponds to one clause, and accumulate the context information for each word of the clause. The hidden state of the jth word in the ith clause $\mathbf{h}_{i,j}$ is obtained based on a bi-directional LSTM. Attention mechanism is then adopt to get a clause representation $\mathbf{s}_i$. Here we omit the details of Bi-LSTM and attention for limited space, readers can refer to Graves et al. (2013) and Bahdanau et al. (2014).

The upper layer consists of two components: one for emotion extraction and another for cause extraction. Each component is a clause-level Bi-LSTM which receives the independent clause representations $[\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{|d|}]$ obtained at the lower layer as inputs. The hidden states of two component Bi-LSTM, $\mathbf{r}_i^e$ and $\mathbf{r}_i^c$, can be viewed as the context-aware representation of clause c_i, and finally feed to the softmax layer for emotion prediction and cause predication:

$$\hat{\mathbf{y}}_i^e = \text{softmax}(\mathbf{W}^e \mathbf{r}_i^e + \mathbf{b}^e), \quad (2)$$

$$\hat{\mathbf{y}}_i^c = \text{softmax}(\mathbf{W}^c \mathbf{r}_i^c + \mathbf{b}^c), \quad (3)$$

where the superscript e and c denotes emotion and cause, respectively.

The loss of the model is a weighted sum of two components:

$$L^p = \lambda L^e + (1 - \lambda) L^c, \quad (4)$$

where L^e and L^c are the cross-entropy error of emotion predication and cause predication respectively, and λ is a tradeoff parameter.

4.1.2 Interactive Multi-task Learning

Till now, two component Bi-LSTM at the upper layer are independent to each other. However, as we have mentioned, the two sub-tasks (emotion extraction and cause extraction) are not mutually independent. On the one hand, providing emotions can help better discover the causes; on the other hand, knowing causes may also help more accurately extract emotions.

Motivated by this, we furthermore propose an interactive multi-task learning network, as an enhanced version of the former one, to capture the correlation between emotion and cause. The structure is shown in Figure 3. It should be noted that the method using emotion extraction to improve cause extraction is called Inter-EC. In addition, we can also use cause extraction to enhance emotion extraction, and call this method Inter-CE. Since Inter-EC and Inter-CE are similar in structure, we only introduce Inter-EC (illustrated in Figure 3 (a)) instead of both.

Compared with Independent Multi-task Learning, the lower layer of Inter-EC is unchanged, and the upper layer consists of two components, which are used to make predictions for emotion extraction task and cause extraction task in an interactive manner. Each component is a clause-level Bi-LSTM followed by a softmax layer.

The first component takes the independent clause representations $[\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{|d|}]$ obtained at the lower layer as inputs for emotion extraction. The hidden state of clause-level Bi-LSTM $\mathbf{r}_i^e$ is used as feature to predict the distribution of the i-th clause $\hat{\mathbf{y}}_i^e$. Then we embed the predicted label of the i-th clause as a vector $\mathbf{Y}_i^e$, which is used for the next component.

Another component takes $(\mathbf{s}_1 \oplus \mathbf{Y}_1^e, \mathbf{s}_2 \oplus \mathbf{Y}_2^e, ..., \mathbf{s}_{|d|} \oplus \mathbf{Y}_{|d|}^e)$ as inputs for cause extraction, where $\oplus$ represents the concatenation operation. The hidden state of clause-level Bi-LSTM $\mathbf{r}_i^c$ is used as feature to predict the distribution of the i-th clause $\hat{\mathbf{y}}_i^c$.

The loss of the model is a weighted sum of two components, which is the same as Equation 4.

4.2 Step 2: Emotion-Cause Pairing and Filtering

In Step 1, we finally obtain a set of emotions $E = \{c_1^e, \cdots, c_m^e\}$ and a set of cause clauses $C = \{c_1^c, \cdots, c_n^c\}$. The goal of Step 2 is then to pair the two sets and construct a set of emotion-

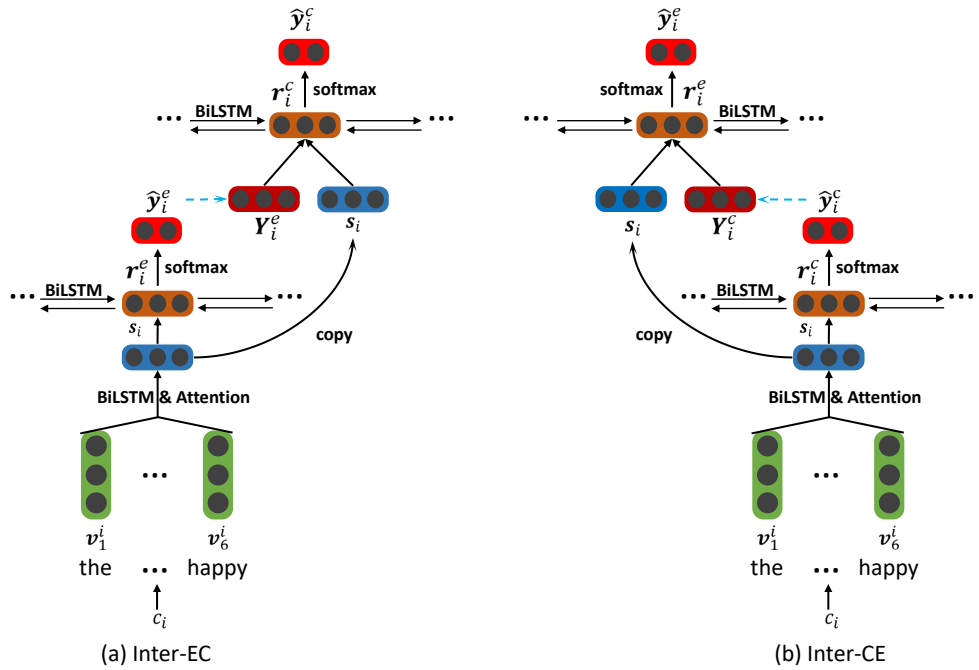


Figure 3: Two Models for Interactive Multi-task Learning: (a) Inter-EC, which uses emotion extraction to improve cause extraction (b) Inter-CE, which uses cause extraction to enhance emotion extraction.

cause pairs with causal relationship.

Firstly, we apply a Cartesian product to E and C, and obtain the set of all possible pairs:

$$P_{all} = \{\cdots, (c_i^e, c_j^c), \cdots\}, \tag{5}$$

Secondly, we represent each pair in P_{all} by a feature vector composed of three kinds of features:

$$\mathbf{x}_{(c_i^e, c_j^c)} = [\mathbf{s}_i^e, \mathbf{s}_j^c, \mathbf{v}^d], \tag{6}$$

where $\mathbf{s}^e$ and $\mathbf{s}^c$ are the representations of the emotion clause and cause clause respectively, and $\mathbf{v}^d$ represents the distances between the two clauses.

A Logistic regression model is then trained to detect for each candidate pair (c_i^e, c_j^c), whether c_i^e and c_j^c have a causal relationship:

$$\hat{y}_{(c_i^e, c_j^c)} \leftarrow \delta(\theta^{\mathrm{T}} \mathbf{x}_{(c_i^e, c_j^c)}), \tag{7}$$

where $\hat{y}_{(c_i^e, c_j^c)} = 1$ denotes that (c_i^e, c_j^c) is a pair with causal relationship, $\hat{y}_{(c_i^e, c_j^c)} = 0$ denotes (c_i^e, c_j^c) is a pair without causal relationship, and $\delta(\cdot)$ is the Sigmoid function. We finally remove the pairs whose $\hat{y}_{(c_i^e, c_j^c)}$ is 0 from P_{all}, and get the final set of emotion-cause pairs.

5 Experiments

5.1 Dataset and Metrics

Since there was no directly available corpus for the ECPE task, we constructed a ECPE corpus based on the benchmark ECE corpus (Gui et al., 2016a), in which each document contains only one emotion and corresponding one or more causes. Documents having two or more emotions are split into several samples such that each contains only one emotion. In order to better meet the ECPE task settings, we merged the documents with the same text content into one document, and labeled each emotion, cause pair in this document. The proportion of documents with different number of emotion-cause pairs in the combined dataset are shown in Table 1.

We stochastically select 90% of the data for training and the remaining 10% for testing. In order to obtain statistically credible results, we repeat the experiments 20 times and report the average result. We use the precision, recall, and F1 score as the metrics for evaluation, which are calculated as follows:

$$P = \frac{\sum correct\_pairs}{\sum proposed\_pairs}, \tag{8}$$

$$R = \frac{\sum correct\_pairs}{\sum annotated\_pairs}, \tag{9}$$

$$F1 = \frac{2 \times P \times R}{P + R}, \tag{10}$$

where $proposed\_pairs$ denotes the number of emotion-cause pairs predicted by the model, $annotated\_pairs$ denotes the total number of

	Number	Percentage
Documents with one emotion-cause pair	1746	89.77%
Documents with two emotion-cause pairs	177	9.10%
Documents with more than two emotion-cause pairs	22	1.13%
All	1945	100%

Table 1: The proportion of documents with different number of emotion-cause pairs in the merged dataset.

	emotion extraction			cause extraction			emotion-cause pair extraction		
	P	R	$F1$	P	R	$F1$	P	R	$F1$
Indep	0.8375	0.8071	0.8210	0.6902	0.5673	0.6205	0.6832	0.5082	0.5818
Inter-CE	**0.8494**	**0.8122**	**0.8300**	0.6809	0.5634	0.6151	**0.6902**	0.5135	0.5901
Inter-EC	0.8364	0.8107	0.8230	**0.7041**	**0.6083**	**0.6507**	0.6721	**0.5705**	**0.6128**

Table 2: Experimental results of all proposed models and variants using precision, recall, and F1-measure as metrics on the ECPE task as well as the two sub-tasks.

emotion-cause pairs that are labeled in the dataset and the $correct\_pairs$ means the number of pairs that are both labeled and predicted as an emotion-cause pair.

In addition, we also evaluate the performance of two sub-tasks: emotion extraction and cause extraction. The precision, recall and F1 score defined in Gui et al. (2016a) are used as the evaluation metrics.

5.2 Experimental Settings

We use word vectors that were pre-trained on the corpora from Chinese Weibo[1] with word2vec (Mikolov et al., 2013) toolkit. The dimension of word embedding is set to 200. The number of hidden units in BiLSTM for all our models is set to 100. All weight matrices and bias are randomly initialized by a uniform distribution U(-0.01, 0.01).

For training details, we use the stochastic gradient descent (SGD) algorithm and Adam update rule with shuffled minibatch. Batch size and learning rate are set to 32 and 0.005, respectively. As for regularization, dropout is applied for word embeddings and the dropout rate is set to 0.8. Besides, we perform L2 constraints over the soft-max parameters and L2-norm regularization is set as 1e-5.[2]

5.3 Evaluation on the ECPE Task

(1) Overall Performance

In Table 2, we report the experimental results of the following three proposed models on three tasks (emotion extraction, cause extraction and emotion-cause pair extraction).

- **Indep**: Indep denotes the method proposed in section 4.1.1. In this method, emotion extraction and cause extraction are independently modeled by two Bi-LSTMs.
- **Inter-CE**: Inter-CE denotes the method proposed in section 4.1.2, where the predictions of cause extraction are used to improve emotion extraction.
- **Inter-EC**: Inter-EC denotes the method proposed in section 4.1.2, where the predictions of emotion extraction are used to enhance cause extraction.

Compared with Indep, Inter-EC gets great improvements on the ECPE task as well as the two sub-tasks. Specifically, we find that the improvements are mainly in the recall rate on the cause extraction task, which finally lead to the great improvement in the recall rate of ECPE. This shows that the predictions of emotion extraction are helpful to cause extraction and proves the effectiveness of Inter-EC. In addition, the performance of emotion extraction also improved, which indicates that the supervision from cause extraction is also beneficial for emotion extraction.

Inter-CE also gets significant improvements on the ECPE task compared to Indep. Specifically, we find that the improvements are mainly in the precision score on the emotion extraction task, which finally lead to the significant improvement in the precision score of ECPE. This shows that the predictions of cause extraction are beneficial to emotion extraction and proves the effectiveness of Inter-CE.

By comparing Inter-EC and Inter-CE, we find that the improvement of Inter-EC is mainly obtained on the cause extraction task, and the im-

[1] http://www.aihuang.org/p/challenge.html

[2] The source code and merged corpus can be obtained at https://github.com/NUSTM/ECPE

	emotion extraction			cause extraction			emotion-cause pair extraction		
	P	R	$F1$	P	R	$F1$	P	R	$F1$
Inter-CE-Bound	#0.9144	#0.8894	#0.9016	#1.0000	#1.0000	#1.0000	#0.8682	#0.8806	#0.8742
Inter-EC-Bound	#1.0000	#1.0000	#1.0000	#0.7842	#0.7116	#0.7452	#0.7610	#0.7084	#0.7328

Table 3: Results of upperbound experiments for Inter-CE and Inter-EC.

	without emotion-cause pair filtering			with emotion-cause pair filtering			
	P	R	$F1$	P	R	$F1$	$keep\_rate$
Indep	0.5894	0.5114	0.5451	0.6832	0.5082	0.5818	0.8507
Inter-CE	0.5883	0.5192	0.5500	**0.6902**	0.5135	0.5901	0.8412
Inter-EC	**0.6019**	**0.5775**	**0.5842**	0.6721	**0.5705**	**0.6128**	0.8889
Inter-CE-Bound	#0.8116	#0.8880	#0.8477	#0.8682	#0.8806	#0.8742	0.9271
Inter-EC-Bound	#0.6941	#0.7118	#0.7018	#0.7610	#0.7084	#0.7328	0.9088

Table 4: Experimental results of all proposed models and variants using precision, recall, and F1-measure as metrics on the ECPE task with or without the pair filter.

provement of Inter-CE is mainly gained on the emotion extraction task. These results are consistent with our intuition that emotion and cause are mutually indicative. In addition, we find that the improvements of Inter-EC on the cause extraction task are much more than the improvement of Inter-CE on the emotion extraction task. We guess that it is because cause extraction is more difficult than emotion extraction, hence there is more room for extra improvement.

(2) Upper-Bound of Emotion and Cause Interaction

In order to further explore the effect of sharing predictions of two sub-tasks, we designed upperbound experiments for Inter-CE and Inter-EC. The results are shown in Table 3.

- **Inter-CE-Bound**: Inter-CE-Bound is a variant of Inter-CE that uses the label of cause extraction to help emotion extraction.
- **Inter-EC-Bound**: Inter-EC-Bound is a variant of Inter-EC that uses the label of emotion extraction to help cause extraction.

The results of Inter-CE-Bound and Inter-EC-Bound are preceded by a "#", indicating that they cannot be compared fairly with other methods because they use annotations. Compared with Indep, the performance of Inter-EC-Bound on cause extraction and the performance of Inter-CE-Bound on emotion extraction both improve greatly. Moreover, the improvement of Inter-EC-Bound on the cause extraction task are much more than the improvement of Inter-CE-Bound on the emotion extraction task. We guess this is because the cause extraction task is more difficult than the emotion extraction task, and there is more room for improvement, which is consistent with previous section.

By comparing the results of Inter-EC-Bound and Inter-EC, we found that although Inter-EC performs better than Indep, it is far poorer than Inter-EC-Bound, which is caused by lots of errors in the predictions of emotion extraction. We can draw the same conclusion when comparing Inter-CE-Bound and Inter-CE.

These experimental results further illustrate that emotion and cause are mutually indicative, and indicate that if we can improve the performance of emotion extraction task, we can get better performance on cause extraction task and vice versa, which finally lead to the improvement on ECPE. But it should be noted it is only an upperbound experiment where the ground-truth of emotion/causes are used to predict each other.

(3) Effect of Emotion-Cause Pair Filtering

In Table 4, we report the emotion-cause pair extraction performance with/without pair filtering. With/Without pair filtering indicates whether we adopt a pair filter after applying a Cartesian product in the second step. $keep\_rate$ indicates the proportion of emotion-cause pairs in P_{all} that are finally retained after pair filtering.

An obvious observation is that the F1 scores of all models on the ECPE task are significantly improved by adopting the pair filter. These results demonstrate the effectiveness of the pair filter. Specifically, by introducing the pair filter, some of the candidate emotion-cause pairs in P_{all} are filtered out, which may result in a decrease in the recall rate and an increase in precision. According to Table 4, the precision scores of almost

	P	R	$F1$
RB	0.6747	0.4287	0.5243
CB	0.2672	0.7130	0.3887
RB+CB+ML	0.5921	0.5307	0.5597
Multi-Kernel	0.6588	0.6927	0.6752
Memnet	0.5922	0.6354	0.6134
ConvMS-Memnet	0.7076	0.6838	0.6955
CANN	07721	0.6891	0.7266
CANN-E	0.4826	0.3160	0.3797
Inter-EC	0.7041	0.6083	0.6507

Table 5: Experimental results of some existing ECE approaches and our model on the ECE task.

all models are greatly improved (more than 7%), in contrast, the recall rates drop very little (less than 1%), which lead to the significant improvement in F1 score.

5.4 Evaluation on the ECE task

In Table 5, we further examine our approach by comparing it with some existing approaches on the traditional ECE task. It should be noted that our Inter-EC model does not use the emotion annotations on the test data.

- **RB** is a rule-based method with manually defined linguistic rules (Lee et al., 2010).
- **CB** is a method based on common-sense knowledge (Russo et al., 2011).
- **RB+CB+ML** (Machine learning method trained from rule-based features and common-sense knowledge base) uses rules and facts in a knowledge base as features and a traditional SVM classifier for classification (Chen et al., 2010).
- **Multi-kernel** uses the multi-kernel method to identify the cause (Gui et al., 2016a).
- **Memnet** denotes a deep memory network proposed by Gui et al. (2017).
- **ConvMS-Memnet** is a convolutional multiple-slot deep memory network proposed by Gui et al. (2017).
- **CANN** denotes a co-attention neural network model proposed in Li et al. (2018).

It can be seen that although our method does not use emotion annotations on the test data, it still achieves comparable results with most of the traditional methods for the ECE task. This indicates that our method can overcome the limitation that emotion annotations must be given at the testing phase in the traditional ECE task, but without reducing the cause extraction performance.

In order to compare with the traditional methods for the ECE task under the same experimental settings, we furthermore implemented a simplification of CANN (CANN-E), which removes the dependency of emotion annotation in the test data.

It is clear that by removing the emotion annotations, the F1 score of CANN drops dramatically (about 34.69%). In contrast, our method does not need the emotion annotations and achieve 65.07% in F1 measure, which significantly outperforms the CANN-E model by 27.1%.

6 Conclusions and Future Work

In this paper, we propose a new task: emotion-cause pair extraction, which aims to extract potential pairs of emotions and corresponding causes in text. To deal with this task, we propose a two-step method, in which we first extract both emotions and causes respectively by multi-task learning, then combine them into pairs by applying Cartesian product, and finally employ a filter to eliminate the false emotion-cause pairs. Based on a benchmark ECE corpus, we construct a corpus suitable for the ECPE task. The experimental results prove the effectiveness of our method.

The two-step strategy may not be a perfect solution to solve the ECPE problem. On the one hand, its goal is not direct. On the other hand, the mistakes made in the first step will affect the results of the second step. In the future work, we will try to build a one-step model that directly extract the emotion-cause pairs in an end-to-end fashion.

Acknowledgments

The work was supported by the Natural Science Foundation of China (No. 61672288), and the Natural Science Foundation of Jiangsu Province for Excellent Young Scholars (No. BK20160085). Rui Xia and Zixiang Ding contributed equally to this paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ying Chen, Wenjun Hou, and Xiyao Cheng. 2018a. Hierarchical convolution neural network for emotion cause detection on microblogs. In *International Conference on Artificial Neural Networks (ICANN)*, pages 115–122.

Ying Chen, Wenjun Hou, Xiyao Cheng, and Shoushan Li. 2018b. Joint learning for emotion classification and emotion cause detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 646–651.

Ying Chen, Sophia Yat Mei Lee, Shoushan Li, and Chu-Ren Huang. 2010. Emotion cause detection with linguistic constructions. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 179–187.

Xiyao Cheng, Ying Chen, Bixiao Cheng, Shoushan Li, and Guodong Zhou. 2017. An emotion cause corpus for chinese microblogs with multiple-user structures. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 17(1):6.

Kai Gao, Hua Xu, and Jiushuo Wang. 2015a. Emotion cause detection for chinese micro-blogs based on ecocc model. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 3–14.

Kai Gao, Hua Xu, and Jiushuo Wang. 2015b. A rule-based approach to emotion cause detection for chinese micro-blogs. *Expert Systems with Applications*, 42(9):4517–4528.

Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. 2015. Detecting emotion stimuli in emotion-bearing sentences. In *International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 152–165.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *IEEE international conference on acoustics, speech and signal processing*. IEEE.

Lin Gui, Jiannan Hu, Yulan He, Ruifeng Xu, Qin Lu, and Jiachen Du. 2017. A question answering approach to emotion cause extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1593–1602.

Lin Gui, Dongyin Wu, Ruifeng Xu, Qin Lu, and Yu Zhou. 2016a. Event-driven emotion cause extraction with corpus construction. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1639–1649.

Lin Gui, Ruifeng Xu, Qin Lu, Dongyin Wu, and Yu Zhou. 2016b. Emotion cause extraction, a challenging task with corpus construction. In *Chinese National Conference on Social Media Processing*, pages 98–109.

Lin Gui, Li Yuan, Ruifeng Xu, Bin Liu, Qin Lu, and Yu Zhou. 2014. Emotion cause detection with linguistic construction in chinese weibo text. In *Natural Language Processing and Chinese Computing (NLPCC)*, pages 457–464.

Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. 2010. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 45–53.

Weiyuan Li and Hua Xu. 2014. Text-based emotion classification using emotion cause extraction. *Expert Systems with Applications*, 41(4):1742–1749.

Xiangju Li, Kaisong Song, Shi Feng, Daling Wang, and Yifei Zhang. 2018. A co-attention neural network model for emotion cause analysis with emotional context awareness. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4752–4757.

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

Alena Neviarouskaya and Masaki Aono. 2013. Extracting causes of emotions from text. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing (IJCNLP)*, pages 932–936.

Irene Russo, Tommaso Caselli, Francesco Rubino, Ester Boldrini, and Patricio Martínez-Barco. 2011. Emocause: an easy-adaptable approach to emotion cause contexts. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, pages 153–160.

Shuangyong Song and Yao Meng. 2015. Detecting concept-level emotion cause in microblogging. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*, pages 119–120.

Bo Xu, Hongfei Lin, Yuan Lin, Yufeng Diao, Liang Yang, and Kan Xu. 2019. Extracting emotion causes using learning to rank methods from an information retrieval perspective. *IEEE Access*.

Ruifeng Xu, Jiannan Hu, Qin Lu, Dongyin Wu, and Lin Gui. 2017. An ensemble approach for emotion cause detection with event extraction and multi-kernel svms. *Tsinghua Science and Technology*, 22(6):646–659.

Shuntaro Yada, Kazushi Ikeda, Keiichiro Hoashi, and Kyo Kageura. 2017. A bootstrap method for automatic rule acquisition on emotion cause extraction. In *IEEE International Conference on Data Mining Workshops*, pages 414–421.

Xinyi Yu, Wenge Rong, Zhuo Zhang, Yuanxin Ouyang, and Zhang Xiong. 2019. Multiple level hierarchical network-based clause selection for emotion cause extraction. *IEEE Access*, 7(1):9071–9079.

Bridging the Gap between Training and Inference for Neural Machine Translation

Wen Zhang[1,2] **Yang Feng**[1,2*] **Fandong Meng**[3] **Di You**[4] **Qun Liu**[5]

[1]Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2]University of Chinese Academy of Sciences, Beijing, China
{zhangwen,fengyang}@ict.ac.cn
[3]Pattern Recognition Center, WeChat AI, Tencent Inc, China
fandongmeng@tencent.com
[4]Worcester Polytechnic Institute, Worcester, MA, USA
dyou@wpi.edu
[5]Huawei Noah's Ark Lab, Hong Kong, China
qun.liu@huawei.com

Abstract

Neural Machine Translation (NMT) generates target words sequentially in the way of predicting the next word conditioned on the context words. At training time, it predicts with the ground truth words as context while at inference it has to generate the entire sequence from scratch. This discrepancy of the fed context leads to error accumulation among the way. Furthermore, word-level training requires strict matching between the generated sequence and the ground truth sequence which leads to overcorrection over different but reasonable translations. In this paper, we address these issues by sampling context words not only from the ground truth sequence but also from the predicted sequence by the model during training, where the predicted sequence is selected with a sentence-level optimum. Experiment results on Chinese→English and WMT'14 English→German translation tasks demonstrate that our approach can achieve significant improvements on multiple datasets.

1 Introduction

Neural Machine Translation has shown promising results and drawn more attention recently. Most NMT models fit in the encoder-decoder framework, including the RNN-based (Sutskever et al., 2014; Bahdanau et al., 2015; Meng and Zhang, 2019), the CNN-based (Gehring et al., 2017) and the attention-based (Vaswani et al., 2017) models, which predict the next word conditioned on the previous context words, deriving a language model over target words. The scenario is at training time the ground truth words are used as context while at inference the entire sequence is generated by the resulting model on its own and hence the previous words generated by the model are fed as context. As a result, the predicted words at training and inference are drawn from different distributions, namely, from the data distribution as opposed to the model distribution. This discrepancy, called *exposure bias* (Ranzato et al., 2015), leads to a gap between training and inference. As the target sequence grows, the errors accumulate among the sequence and the model has to predict under the condition it has never met at training time.

Intuitively, to address this problem, the model should be trained to predict under the same condition it will face at inference. Inspired by DATA AS DEMONSTRATOR (DAD) (Venkatraman et al., 2015), feeding as context both ground truth words and the predicted words during training can be a solution. NMT models usually optimize the cross-entropy loss which requires a strict pairwise matching at the word level between the predicted sequence and the ground truth sequence. Once the model generates a word deviating from the ground truth sequence, the cross-entropy loss will correct the error immediately and draw the remaining generation back to the ground truth sequence. However, this causes a new problem. A sentence usually has multiple reasonable translations and it cannot be said that the model makes a mistake even if it generates a word different from the ground truth word. For example,

reference:	We should comply with the rule.
cand1:	We should abide with the rule.
cand2:	We should abide by the law.
cand3:	We should abide by the rule.

*Corresponding author.

once the model generates "abide" as the third target word, the cross-entropy loss would force the model to generate "with" as the fourth word (as *cand1*) so as to produce larger sentence-level likelihood and be in line with the reference, although "by" is the right choice. Then, "with" will be fed as context to generate "the rule", as a result, the model is taught to generate "abide with the rule" which actually is wrong. The translation *cand1* can be treated as *overcorrection* phenomenon. Another potential error is that even the model predicts the right word "by" following "abide", when generating subsequent translation, it may produce "the law" improperly by feeding "by" (as *cand2*). Assume the references and the training criterion let the model memorize the pattern of the phrase "the rule" always following the word "with", to help the model recover from the two kinds of errors and create the correct translation like *cand3*, we should feed "with" as context rather than "by" even when the previous predicted phrase is "abide by". We refer to this solution as *Overcorrection Recovery* (*OR*).

In this paper, we present a method to bridge the gap between training and inference and improve the overcorrection recovery capability of NMT. Our method first selects *oracle* words from its predicted words and then samples as context from the oracle words and ground truth words. Meanwhile, the oracle words are selected not only with a word-by-word greedy search but also with a sentence-level evaluation, e.g. BLEU, which allows greater flexibility under the pairwise matching restriction of cross-entropy. At the beginning of training, the model selects as context ground truth words at a greater probability. As the model converges gradually, oracle words are chosen as context more often. In this way, the training process changes from a fully guided scheme towards a less guided scheme. Under this mechanism, the model has the chance to learn to handle the mistakes made at inference and also has the ability to recover from overcorrection over alternative translations. We verify our approach on both the RNNsearch model and the stronger Transformer model. The results show that our approach can significantly improve the performance on both models.

2 RNN-based NMT Model

Our method can be applied in a variety of NMT models. Without loss of generality, we take the RNN-based NMT (Bahdanau et al., 2015) as an example to introduce our method. Assume the source sequence and the observed translation are $\mathbf{x} = \{x_1, \cdots, x_{|\mathbf{x}|}\}$ and $\mathbf{y}^* = \{y_1^*, \cdots, y_{|\mathbf{y}^*|}^*\}$.

Encoder. A bidirectional Gated Recurrent Unit (GRU) (Cho et al., 2014) is used to acquire two sequences of hidden states, the annotation of x_i is $h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i]$. Note that e_{x_i} is employed to represent the embedding vector of the word x_i.

$$\overrightarrow{h}_i = \mathbf{GRU}(e_{x_i}, \overrightarrow{h}_{i-1}) \tag{1}$$

$$\overleftarrow{h}_i = \mathbf{GRU}(e_{x_i}, \overleftarrow{h}_{i+1}) \tag{2}$$

Attention. The attention is designed to extract source information (called source context vector). At the j-th step, the relevance between the target word y_j^* and the i-th source word is evaluated and normalized over the source sequence

$$r_{ij} = \mathbf{v}_a^T \tanh\left(\mathbf{W}_a s_{j-1} + \mathbf{U}_a h_i\right) \tag{3}$$

$$\alpha_{ij} = \frac{\exp\left(r_{ij}\right)}{\sum_{i'=1}^{|\mathbf{x}|} \exp\left(r_{i'j}\right)} \tag{4}$$

The source context vector is the weighted sum of all source annotations and can be calculated by

$$c_j = \sum\nolimits_{i=1}^{|\mathbf{x}|} \alpha_{ij} h_i \tag{5}$$

Decoder. The decoder employs a variant of GRU to unroll the target information. At the j-th step, the target hidden state s_j is given by

$$s_j = \mathbf{GRU}(e_{y_{j-1}^*}, s_{j-1}, c_j) \tag{6}$$

The probability distribution P_j over all the words in the target vocabulary is produced conditioned on the embedding of the previous ground truth word, the source context vector and the hidden state

$$t_j = g\left(e_{y_{j-1}^*}, c_j, s_j\right) \tag{7}$$

$$o_j = \mathbf{W}_o t_j \tag{8}$$

$$P_j = \mathrm{softmax}\left(o_j\right) \tag{9}$$

where g stands for a linear transformation, $\mathbf{W}_o$ is used to map t_j to o_j so that each target word has one corresponding dimension in o_j.

3 Approach

The main framework (as shown in Figure 1) of our method is to feed as context either the ground truth words or the previous predicted words, i.e. *oracle*

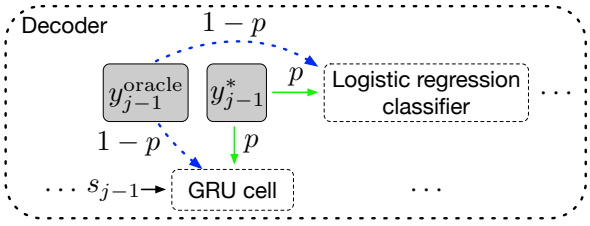


Figure 1: The architecture of our method.

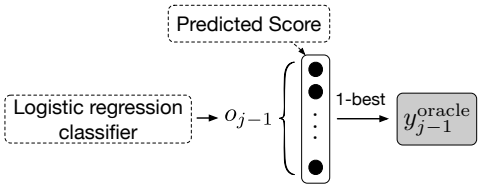


Figure 2: Word-level oracle without noise.

words, with a certain probability. This potentially can reduce the gap between training and inference by training the model to handle the situation which will appear during test time. We will introduce two methods to select the oracle words. One method is to select the oracle words at the word level with a greedy search algorithm, and another is to select a oracle sequence at the sentence-level optimum. The sentence-level oracle provides an option of n-gram matching with the ground truth sequence and hence inherently has the ability of recovering from overcorrection for the alternative context. To predict the j-th target word y_j, the following steps are involved in our approach:

1. Select an oracle word y_{j-1}^{oracle} (at word level or sentence level) at the $\{j-1\}$-th step. (Section **Oracle Word Selection**)

2. Sample from the ground truth word y_{j-1}^{*} with a probability of p or from the oracle word y_{j-1}^{oracle} with a probability of $1-p$. (Section **Sampling with Decay**)

3. Use the sampled word as y_{j-1} and replace the y_{j-1}^{*} in Equation (6) and (7) with y_{j-1}, then perform the following prediction of the attention-based NMT.

3.1 Oracle Word Selection

Generally, at the j-th step, the NMT model needs the ground truth word y_{j-1}^{*} as the context word to predict y_j, thus, we could select an oracle word y_{j-1}^{oracle} to simulate the context word. The oracle word should be a word similar to the ground truth or a synonym. Using different strategies will produce a different oracle word y_{j-1}^{oracle}. One option is that word-level greedy search could be employed to output the oracle word of each step, which is called *Word-level Oracle* (called WO). Besides, we can further optimize the oracle by enlarging the search space with beam search and then re-ranking the candidate translations with a sentence-level metric, e.g. BLEU (Papineni et al., 2002), GLEU (Wu et al., 2016), ROUGE (Lin, 2004), etc, the selected translation is called *oracle sentence*, the words in the translation are *Sentence-level Oracle* (denoted as SO).

Word-Level Oracle

For the $\{j-1\}$-th decoding step, the direct way to select the word-level oracle is to pick the word with the highest probability from the word distribution P_{j-1} drawn by Equation (9), which is shown in Figure 2. The predicted score in o_{j-1} is the value before the $\mathrm{softmax}$ operation. In practice, we can acquire more robust word-level oracles by introducing the *Gumbel-Max* technique (Gumbel, 1954; Maddison et al., 2014), which provides a simple and efficient way to sample from a categorical distribution.

The Gumbel noise, treated as a form of regularization, is added to o_{j-1} in Equation (8), as shown in Figure 3, then $\mathrm{softmax}$ function is performed, the word distribution of y_{j-1} is approximated by

$$\eta = -\log\left(-\log u\right) \tag{10}$$

$$\tilde{o}_{j-1} = \left(o_{j-1} + \eta\right)/\tau \tag{11}$$

$$\tilde{P}_{j-1} = \mathrm{softmax}\left(\tilde{o}_{j-1}\right) \tag{12}$$

where η is the Gumbel noise calculated from a uniform random variable $u \sim \mathcal{U}(0,1)$, τ is temperature. As τ approaches 0, the $\mathrm{softmax}$ function is similar to the argmax operation, and it becomes uniform distribution gradually when $\tau \to \infty$. Similarly, according to $\tilde{P}_{j-1}$, the 1-best word is selected as the word-level oracle word

$$y_{j-1}^{\text{oracle}} = y_{j-1}^{\text{WO}} = \mathrm{argmax}\left(\tilde{P}_{j-1}\right) \tag{13}$$

Note that the Gumbel noise is just used to select the oracle and it does not affect the loss function for training.

Sentence-Level Oracle

The sentence-level oracle is employed to allow for more flexible translation with n-gram matching required by a sentence-level metric. In this paper,

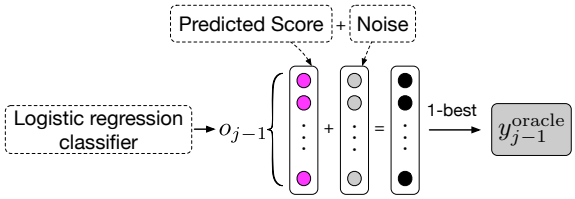


Figure 3: Word-level oracle with Gumbel noise.

we employ BLEU as the sentence-level metric. To select the sentence-level oracles, we first perform beam search for all sentences in each batch, assuming beam size is k, and get k-best candidate translations. In the process of beam search, we also could apply the Gumbel noise for each word generation. We then evaluate each translation by calculating its BLEU score with the ground truth sequence, and use the translation with the highest BLEU score as the *oracle sentence*. We denote it as $\mathbf{y}^{\text{S}} = (y_1^{\text{S}}, ..., y_{|\mathbf{y}^{\text{S}}|}^{\text{S}})$, then at the j-th decoding step, we define the sentence-level oracle word as

$$y_{j-1}^{\text{oracle}} = y_{j-1}^{\text{SO}} = y_{j-1}^{\text{S}} \tag{14}$$

But a problem comes with sentence-level oracle. As the model samples from ground truth word and the sentence-level oracle word at each step, the two sequences should have the same number of words. However we can not assure this with the naive beam search decoding algorithm. Based on the above problem, we introduce *force decoding* to make sure the two sequences have the same length.

Force Decoding. As the length of the ground truth sequence is $|\mathbf{y}^*|$, the goal of force decoding is to generate a sequence with $|\mathbf{y}^*|$ words followed by a special end-of-sentence (EOS) symbol. Therefore, in beam search, once a candidate translation tends to end with EOS when it is shorter or longer than $|\mathbf{y}^*|$, we will force it to generate $|\mathbf{y}^*|$ words, that is,

- If the candidate translation gets a word distribution P_j at the j-th step where $j \leqslant |\mathbf{y}^*|$ and EOS is the top first word in P_j, then we select the top second word in P_j as the j-th word of this candidate translation.
- If the candidate translation gets a word distribution $P_{|\mathbf{y}^*|+1}$ at the $\{|\mathbf{y}^*|+1\}$-th step where EOS is not the top first word in $P_{|\mathbf{y}^*|+1}$, then we select EOS as the $\{|\mathbf{y}^*|+1\}$-th word of this candidate translation.

In this way, we can make sure that all the k candidate translations have $|\mathbf{y}^*|$ words, then re-rank the k candidates according to BLEU score and select the top first as the oracle sentence. For adding Gumbel noise into the sentence-level oracle selection, we replace the P_j with $\tilde{P}_j$ at the j-th decoding step during force decoding.

3.2 Sampling with Decay

In our method, we employ a sampling mechanism to randomly select the ground truth word y_{j-1}^* or the oracle word y_{j-1}^{oracle} as y_{j-1}. At the beginning of training, as the model is not well trained, using y_{j-1}^{oracle} as y_{j-1} too often would lead to very slow convergence, even being trapped into local optimum. On the other hand, at the end of training, if the context y_{j-1} is still selected from the ground truth word y_{j-1}^* at a large probability, the model is not fully exposed to the circumstance which it has to confront at inference and hence can not know how to act in the situation at inference. In this sense, the probability p of selecting from the ground truth word can not be fixed, but has to decrease progressively as the training advances. At the beginning, p=1, which means the model is trained entirely based on the ground truth words. As the model converges gradually, the model selects from the oracle words more often.

Borrowing ideas from but being different from Bengio et al. (2015) which used a schedule to decrease p as a function of the index of mini-batch, we define p with a decay function dependent on the index of training epochs e (starting from 0)

$$p = \frac{\mu}{\mu + \exp\left(e/\mu\right)} \tag{15}$$

where μ is a hyper-parameter. The function is strictly monotone decreasing. As the training proceeds, the probability p of feeding ground truth words decreases gradually.

3.3 Training

After selecting y_{j-1} by using the above method, we can get the word distribution of y_j according to Equation (6), (7), (8) and (9). We do not add the Gumbel noise to the distribution when calculating loss for training. The objective is to maximize the probability of the ground truth sequence based on maximum likelihood estimation (MLE). Thus following loss function is minimized:

$$\mathcal{L}\left(\theta\right) = -\sum\nolimits_{n=1}^{N}\sum\nolimits_{j=1}^{|\mathbf{y}^n|}\log P_j^n\left[y_j^n\right] \tag{16}$$

where N is the number of sentence pairs in the training data, $|\mathbf{y}^n|$ indicates the length of the n-th

ground truth sentence, P_j^n refers to the predicted probability distribution at the j-th step for the n-th sentence, hence $P_j^n\left[y_j^n\right]$ is the probability of generating the ground truth word y_j^n at the j-th step.

4 Related Work

Some other researchers have noticed the problem of exposure bias in NMT and tried to solve it. Venkatraman et al. (2015) proposed DATA AS DEMONSTRATOR (DAD) which initialized the training examples as the paired two adjacent ground truth words and at each step added the predicted word paired with the next ground truth word as a new training example. Bengio et al. (2015) further developed the method by sampling as context from the previous ground truth word and the previous predicted word with a changing probability, not treating them equally in the whole training process. This is similar to our method, but they do not include the sentence-level oracle to relieve the overcorrection problem and neither the noise perturbations on the predicted distribution.

Another direction of attempts is the sentence-level training with the thinking that the sentence-level metric, e.g., BLEU, brings a certain degree of flexibility for generation and hence is more robust to mitigate the exposure bias problem. To avoid the problem of exposure bias, Ranzato et al. (2015) presented a novel algorithm Mixed Incremental Cross-Entropy Reinforce (MIXER) for sequence-level training, which directly optimized the sentence-level BLEU used at inference. Shen et al. (2016) introduced the Minimum Risk Training (MRT) into the end-to-end NMT model, which optimized model parameters by minimizing directly the expected loss with respect to arbitrary evaluation metrics, e.g., sentence-level BLEU. Shao et al. (2018) proposed to eliminate the exposure bias through a probabilistic n-gram matching objective, which trains NMT NMT under the greedy decoding strategy.

5 Experiments

We carry out experiments on the NIST Chinese→English (Zh→En) and the WMT'14 English→German (En→De) translation tasks.

5.1 Settings

For Zh→En, the training dataset consists of 1.25M sentence pairs extracted from LDC corpora[1]. We choose the NIST 2002 (MT02) dataset as the validation set, which has 878 sentences, and the NIST 2003 (MT03), NIST 2004 (MT04), NIST 2005 (MT05) and NIST 2006 (MT06) datasets as the test sets, which contain 919, 1788, 1082 and 1664 sentences respectively. For En→De, we perform our experiments on the corpus provided by WMT'14, which contains 4.5M sentence pairs[2]. We use the newstest2013 as the validation set, and the newstest2014 as the test sets, which containing 3003 and 2737 sentences respectively. We measure the translation quality with BLEU scores (Papineni et al., 2002). For Zh→En, case-insensitive BLEU score is calculated by using the *mteval-v11b.pl* script. For En→De, we tokenize the references and evaluate the performance with case-sensitive BLEU score by the *multi-bleu.pl* script. The metrics are exactly the same as in previous work. Besides, we make statistical significance test according to the method of Collins et al. (2005).

In training the NMT model, we limit the source and target vocabulary to the most frequent 30K words for both sides in the Zh→En translation task, covering approximately 97.7% and 99.3% words of two corpus respectively. For the En→De translation task, sentences are encoded using byte-pair encoding (BPE) (Sennrich et al., 2016) with $37k$ merging operations for both source and target languages, which have vocabularies of 39418 and 40274 tokens respectively. We limit the length of sentences in the training datasets to 50 words for Zh→En and 128 subwords for En→De. For RNNSearch model, the dimension of word embedding and hidden layer is 512, and the beam size in testing is 10. All parameters are initialized by the uniform distribution over $[-0.1, 0.1]$. The mini-batch stochastic gradient descent (SGD) algorithm is employed to train the model parameters with batch size setting to 80. Moreover, the learning rate is adjusted by adadelta optimizer (Zeiler, 2012) with ρ=0.95 and ϵ=1e-6. Dropout is applied on the output layer with dropout rate being 0.5. For Transformer model, we train base model with

[1]These sentence pairs are mainly extracted from LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06

[2]http://www.statmt.org/wmt14/translation-task.html

Systems	Architecture	MT03	MT04	MT05	MT06	Average
Existing end-to-end NMT systems						
Tu et al. (2016)	Coverage	33.69	38.05	35.01	34.83	35.40
Shen et al. (2016)	MRT	37.41	39.87	37.45	36.80	37.88
Zhang et al. (2017)	Distortion	37.93	40.40	36.81	35.77	37.73
Our end-to-end NMT systems						
this work	RNNsearch	37.93	40.53	36.65	35.80	37.73
	+ SS-NMT	38.82	41.68	37.28	37.98	38.94
	+ MIXER	38.70	40.81	37.59	38.38	38.87
	+ OR-NMT	**40.40**‡†*	**42.63**‡†*	**38.87**‡†*	**38.44**‡	**40.09**
	Transformer	46.89	47.88	47.40	46.66	47.21
	+ word oracle	47.42	48.34	47.89	47.34	47.75
	+ sentence oracle	**48.31***	**49.40***	**48.72***	**48.45***	**48.72**

Table 1: Case-insensitive BLEU scores (%) on Zh→En translation task. "‡", "†", "⋆" and "∗" indicate statistically significant difference (p<0.01) from RNNsearch, SS-NMT, MIXER and Transformer, respectively.

default settings (fairseq[3]).

5.2 Systems

The following systems are involved:

RNNsearch: Our implementation of an improved model as described in Section 2, where the decoder employs two GRUs and an attention. Specifically, Equation 6 is substituted with:

$$\tilde{s}_j = \mathbf{GRU}_1(e_{y^*_{j-1}}, s_{j-1}) \quad (17)$$

$$s_j = \mathbf{GRU}_2(c_j, \tilde{s}_j) \quad (18)$$

Besides, in Equation 3, s_{j-1} is replaced with $\tilde{s}_{j-1}$.

SS-NMT: Our implementation of the scheduled sampling (SS) method (Bengio et al., 2015) on the basis of the RNNsearch. The decay scheme is the same as Equation 15 in our approach.

MIXER: Our implementation of the mixed incremental cross-entropy reinforce (Ranzato et al., 2015), where the sentence-level metric is BLEU and the average reward is acquired according to its offline method with a 1-layer linear regressor.

OR-NMT: Based on the RNNsearch, we introduced the word-level oracles, sentence-level oracles and the Gumbel noises to enhance the overcorrection recovery capacity. For the sentence-level oracle selection, we set the beam size to be 3, set τ=0.5 in Equation (11) and μ=12 for the decay function in Equation (15). OR-NMT is the abbreviation of NMT with Overcorrection Recovery.

[3] https://github.com/pytorch/fairseq

5.3 Results on Zh→En Translation

We verify our method on two baseline models with the NIST Zh→En datasets in this section.

Results on the RNNsearch

As shown in Table 1, Tu et al. (2016) propose to model coverage in RNN-based NMT to improve the adequacy of translations. Shen et al. (2016) propose minimum risk training (MRT) for NMT to directly optimize model parameters with respect to BLEU scores. Zhang et al. (2017) model distortion to enhance the attention model. Compared with them, our baseline system RNNsearch 1) outperforms previous shallow RNN-based NMT system equipped with the coverage model (Tu et al., 2016); and 2) achieves competitive performance with the MRT (Shen et al., 2016) and the Distortion (Zhang et al., 2017) on the same datasets. We hope that the strong shallow baseline system used in this work makes the evaluation convincing.

We also compare with the other two related methods that aim at solving the exposure bias problem, including the scheduled sampling (Bengio et al., 2015) (SS-NMT) and the sentence-level training (Ranzato et al., 2015) (MIXER). From Table 1, we can see that both SS-NMT and MIXER can achieve improvements by taking measures to mitigate the exposure bias. While our approach OR-NMT can outperform the baseline system RNNsearch and the competitive comparison systems by directly incorporate the sentence-level oracle and noise perturbations for relieving the overcorrection problem. Particularly, our OR-NMT significantly outperforms the RNNsearch by +2.36 BLEU points averagely on four test datasets. Comparing with the two related models,

Systems	Average
RNNsearch	37.73
+ word oracle	38.94
+ noise	39.50
+ sentence oracle	39.56
+ noise	**40.09**

Table 2: Factor analysis on Zh→En translation, the results are average BLEU scores on MT03∼06 datasets.

our approach further gives a significant improvements on most test sets and achieves improvement by about +1.2 BLEU points on average.

Results on the Transformer

The methods we propose can also be adapted to the stronger Transformer model. The evaluated results are listed in Table 1. Our word-level method can improve the base model by +0.54 BLEU points on average, and the sentence-level method can further bring in +1.0 BLEU points improvement.

5.4 Factor Analysis

We propose several strategies to improve the performance of approach on relieving the overcorrection problem, including utilizing the word-level oracle, the sentence-level oracle, and incorporating the Gumbel noise for oracle selection. To investigate the influence of these factors, we conduct the experiments and list the results in Table 2.

When only employing the word-level oracle, the translation performance was improved by +1.21 BLEU points, this indicates that feeding predicted words as context can mitigate exposure bias. When employing the sentence-level oracle, we can further achieve +0.62 BLEU points improvement. It shows that the sentence-level oracle performs better than the word-level oracle in terms of BLEU. We conjecture that the superiority may come from a greater flexibility for word generation which can mitigate the problem of overcorrection. By incorporating the Gumbel noise during the generation of the word-level and sentence-level oracle words, the BLEU score are further improved by 0.56 and 0.53 respectively. This indicates Gumbel noise can help the selection of each oracle word, which is consistent with our claim that Gumbel-Max provides a efficient and robust way to sample from a categorical distribution.

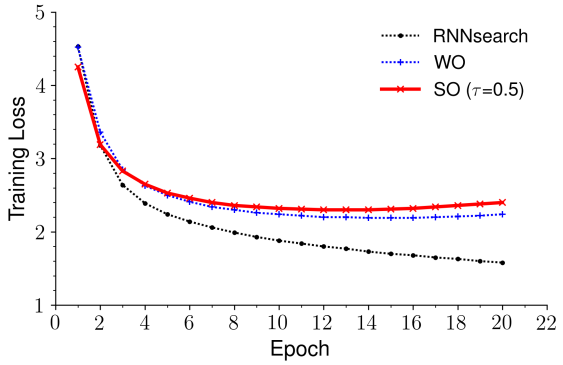


Figure 4: Training loss curves on Zh→En translation with different factors. The black, blue and red colors represent the RNNsearch, RNNsearch with word-level oracle and RNNsearch with sentence-level oracle systems respectively.

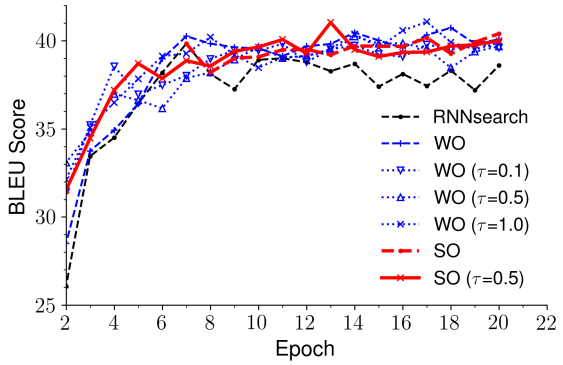


Figure 5: Trends of BLEU scores on the validation set with different factors on the Zh→En translation task.

5.5 About Convergence

In this section, we analyze the influence of different factors for the convergence. Figure 4 gives the training loss curves of the RNNsearch, word-level oracle (WO) without noise and sentence-level oracle (SO) with noise. In training, BLEU score on the validation set is used to select the best model, a detailed comparison among the BLEU score curves under different factors is shown in Figure 5. RNNsearch converges fast and achieves the best result at the 7-th epoch, while the training loss continues to decline after the 7-th epoch until the end. Thus, the training of RNNsearch may encounter the overfitting problem. Figure 4 and 5 also reveal that, integrating the oracle sampling and the Gumbel noise leads to a little slower convergence and the training loss does not keep decreasing after the best results appear on the validation set. This is consistent with our intuition that oracle sampling and noises can avoid overfit-

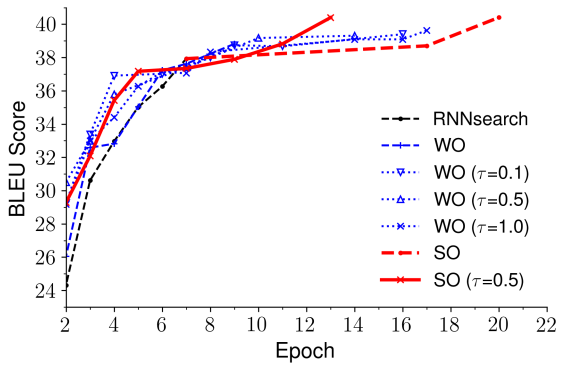


Figure 6: Trends of BLEU scores on the MT03 test set with different factors on the Zh→En translation task.

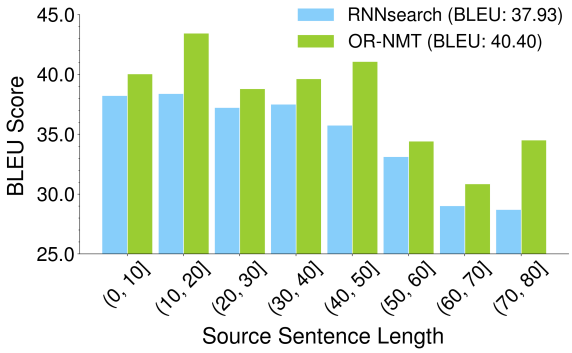


Figure 7: Performance comparison on the MT03 test set with respect to the different lengths of source sentences on the Zh→En translation task.

ting despite needs a longer time to converge.

Figure 6 shows the BLEU scores curves on the MT03 test set under different factors[4]. When sampling oracles with noise (τ=0.5) on the sentence level, we obtain the best model. Without noise, our system converges to a lower BLEU score. This can be understood easily that using its own results repeatedly during training without any regularization will lead to overfitting and quick convergence. In this sense, our method benefits from the sentence-level sampling and Gumbel noise.

5.6 About Length

Figure 7 shows the BLEU scores of generated translations on the MT03 test set with respect to the lengths of the source sentences. In particular, we split the translations for the MT03 test set into different bins according to the length of source sentences, then test the BLEU scores for translations in each bin separately with the results reported in Figure 7. Our approach can achieve big improvements over the baseline system in all bins, especially in the bins (10,20], (40,50] and (70,80] of the super-long sentences. The cross-entropy loss requires that the predicted sequence is exactly the same as the ground truth sequence which is more difficult to achieve for long sentences, while our sentence-level oracle can help recover from this kind of overcorrection.

5.7 Effect on Exposure Bias

To validate whether the improvements is mainly obtained by addressing the exposure bias problem, we randomly select 1K sentence pairs from the Zh→En training data, and use the pre-trained RNNSearch model and proposed model to decode the source sentences. The BLEU score of RNNSearch model was 24.87, while our model produced +2.18 points. We then count the ground truth words whose probabilities in the predicted distributions produced by our model are greater than those produced by the baseline model, and mark the number as $\mathcal{N}$. There are totally $28,266$ gold words in the references, and $\mathcal{N}$=$18,391$. The proportion is $18,391/28,266$=65.06%, which could verify the improvements are mainly obtained by addressing the exposure bias problem.

5.8 Results on En→De Translation

Systems	newstest2014
RNNsearch	25.82
+ SS-NMT	26.50
+ MIXER	26.76
+ OR-NMT	**27.41**‡
Transformer (base)	27.34
+ SS-NMT	28.05
+ MIXER	27.98
+ OR-NMT	**28.65**‡

Table 3: Case-sensitive BLEU scores (%) on En→De task. The "‡" indicates the results are significantly better ($p<0.01$) than RNNsearch and Transformer.

We also evaluate our approach on the WMT'14 benchmarks on the En→De translation task. From the results listed in Table 3, we conclude that the proposed method significantly outperforms the competitive baseline model as well as related approaches. Similar with results on the Zh→En task, both scheduled sampling and MIXER could improve the two baseline systems. Our method im-

[4]Note that the "SO" model without noise is trained based on the pre-trained RNNsearch model (as shown by the red dashed lines in Figure 5 and 6).

proves the RNNSearch and Transformer baseline models by +1.59 and +1.31 BLEU points respectively. These results demonstrate that our model works well across different language pairs.

6 Conclusion

The end-to-end NMT model generates a translation word by word with the ground truth words as context at training time as opposed to the previous words generated by the model as context at inference. To mitigate the discrepancy between training and inference, when predicting one word, we feed as context either the ground truth word or the previous predicted word with a sampling scheme. The predicted words, referred to as oracle words, can be generated with the word-level or sentence-level optimization. Compared to word-level oracle, sentence-level oracle can further equip the model with the ability of overcorrection recovery. To make the model fully exposed to the circumstance at reference, we sample the context word with decay from the ground truth words. We verified the effectiveness of our method with two strong baseline models and related works on the real translation tasks, achieved significant improvement on all the datasets. We also conclude that the sentence-level oracle show superiority over the word-level oracle.

Acknowledgments

We thank the three anonymous reviewers for their valuable suggestions. This work was supported by National Natural Science Foundation of China (NO. 61662077, NO. 61876174) and National Key R&D Program of China (NO. YS2017YFGH001428).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR 2015*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1171–1179. Curran Associates, Inc.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia. PMLR.

Emil Julius Gumbel. 1954. Statistical theory of extreme valuse and some practical applications. *Nat. Bur. Standards Appl. Math. Ser. 33*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chris J Maddison, Daniel Tarlow, and Tom Minka. 2014. A* sampling. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3086–3094. Curran Associates, Inc.

Fandong Meng and Jinchao Zhang. 2019. Dtmt: A novel deep transition architecture for neural machine translation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, AAAI'19. AAAI Press.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Chenze Shao, Xilin Chen, and Yang Feng. 2018. Greedy search with probabilistic n-gram matching for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4778–4784.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1683–1692.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of ACL*.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. 2015. Improving multi-step prediction of learned time series models. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 3024–3030. AAAI Press.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Jinchao Zhang, Mingxuan Wang, Qun Liu, and Jie Zhou. 2017. Incorporating word reordering knowledge into attention-based neural machine translation. In *Proceedings of ACL*.

Zero-shot Word Sense Disambiguation using Sense Definition Embeddings

Sawan Kumar[1] **Sharmistha Jat**[1] **Karan Saxena**[2,*] **Partha Talukdar**[1]
[1] Indian Institute of Science, Bangalore
[2] Carnegie Mellon University, Pittsburgh
{sawankumar,sharmisthaj,ppt}@iisc.ac.in, karansax@cs.cmu.edu

Abstract

Word Sense Disambiguation (WSD) is a long-standing but open problem in Natural Language Processing (NLP). WSD corpora are typically small in size, owing to an expensive annotation process. Current supervised WSD methods treat senses as discrete labels and also resort to predicting the Most-Frequent-Sense (MFS) for words unseen during training. This leads to poor performance on rare and unseen senses. To overcome this challenge, we propose Extended WSD Incorporating Sense Embeddings (EWISE), a supervised model to perform WSD by predicting over a continuous sense embedding space as opposed to a discrete label space. This allows EWISE to generalize over both seen and unseen senses, thus achieving generalized zero-shot learning. To obtain target sense embeddings, EWISE utilizes sense definitions. EWISE learns a novel sentence encoder for sense definitions by using WordNet relations and also ConvE, a recently proposed knowledge graph embedding method. We also compare EWISE against other sentence encoders pretrained on large corpora to generate definition embeddings. EWISE achieves new state-of-the-art WSD performance.

1 Introduction

Word Sense Disambiguation (WSD) is an important task in Natural Language Processing (NLP) (Navigli, 2009). The task is to associate a word in text to its correct sense, where the set of possible senses for the word is assumed to be known a priori. Consider the noun "tie" and the following examples of its usage (Miller, 1995).

- "*he wore a vest and tie*"
- "*their record was 3 wins, 6 losses and a tie*"

It is clear that the implied sense of the word "tie" is very different in the two cases. The word is associated with *"neckwear consisting of a long narrow piece of material"* in the first example, and with *"the finish of a contest in which the winner is undecided"* in the second. The goal of WSD is to predict the right sense, given a word and its context.

WSD has been shown to be useful for popular NLP tasks such as machine translation (Neale et al., 2016; Pu et al., 2018), information extraction (Zhong and Ng, 2012; Delli Bovi et al., 2015) and question answering (Ramakrishnan et al., 2003). The task of WSD can also be viewed as an intrinsic evaluation benchmark for the semantics learned by sentence comprehension models. WSD remains an open problem despite a long history of research. In this work, we study the all-words WSD task, where the goal is to disambiguate all ambiguous words in a corpus.

Supervised (Zhong and Ng, 2010; Iacobacci et al., 2016; Melamud et al., 2016) and semi-supervised approaches (Taghipour and Ng, 2015; Yuan et al., 2016) to WSD treat the target senses as discrete labels. Treating senses as discrete labels limits the generalization capability of these models for senses which occur infrequently in the training data. Further, for disambiguation of words not seen during training, these methods fall back on using a Most-Frequent-Sense (MFS) strategy, obtained from an external resource such as WordNet (Miller, 1995). To address these concerns, unsupervised knowledge-based (KB) approaches have been introduced, which rely solely on lexical resources (e.g., WordNet). KB methods include approaches based on context-definition overlap (Lesk, 1986; Basile et al., 2014), or on the structural properties of the lexical resource (Moro et al., 2014; Weissenborn et al., 2015; Chaplot et al., 2015; Chaplot and Salakhutdinov, 2018;

[*] Work done as a Research Assistant at Indian Institute of Science, Bangalore.

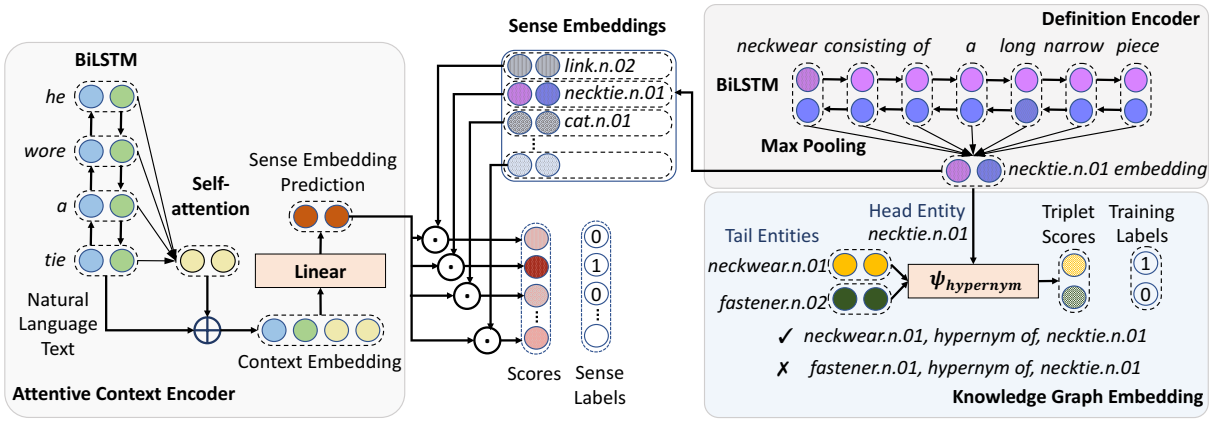


Figure 1: *Overview of WSD in EWISE*: A sequence of input tokens is encoded into context-aware embeddings using a BiLSTM and a self-attention layer ($\oplus$ indicates concatenation). The context-aware embeddings are then projected on to the space of sense embeddings. The score for each sense in the sense inventory is obtained using a dot product (indicated by $\odot$) of the sense embedding with the projected word embedding. Please see Section 4.2 for details on the context encoding and training of the context encoder. The sense embedding for each sense in the inventory is generated using a BiLSTM-Max definition encoder. The encoder is learnt using the training signal present in WordNet Graph. An example signal with hypernym relation is depicted. Please see Section 4.3 for details on learning sense embeddings.

Tripodi and Pelillo, 2017).

While knowledge-based approaches offer a way to disambiguate rare and unseen words into potentially rare senses, supervised methods consistently outperform these methods in the general setting where inference is to be carried over both frequently occurring and rare words. Recently, Raganato et al. (2017b) posed WSD as a neural sequence labeling task, further improving the state-of-the-art. Yet, owing to an expensive annotation process (Lopez de Lacalle and Agirre, 2015), there is a scarcity of sense-annotated data thereby limiting the generalization ability of supervised methods. While there has been recent interest in incorporating definitions (glosses) to overcome the supervision bottleneck for WSD (Luo et al., 2018b,a), these methods are still limited due to their treatment of senses as discrete labels.

Our hypothesis is that supervised methods can leverage lexical resources to improve on WSD for both observed and unobserved words and senses. We propose **Extended WSD Incorporating Sense Embeddings** (**EWISE**). Instead of learning a model to choose between discrete labels, EWISE learns a continuous space of sense embeddings as target. This enables generalized zero-shot learning, i.e., the ability to recognize instances of seen as well as unseen senses. EWISE utilizes sense definitions and additional information from lexical resources. We believe that natural language information manually encoded into definitions contains a rich source of information for representation learning of senses.

To obtain definition embeddings, we propose a novel learning framework which leverages recently successful Knowledge Graph (KG) embedding methods (Bordes et al., 2013; Dettmers et al., 2018). We also compare against sentence encoders pretrained on large corpora.

In summary, we make the following contributions in this work.

- We propose EWISE, a principled framework to learn from a combination of sense-annotated data, dictionary definitions and lexical knowledge bases.
- We propose the use of sense embeddings instead of discrete labels as the targets for supervised WSD, enabling generalized zero-shot learning.
- Through extensive evaluation, we demonstrate the effectiveness of EWISE over state-of-the-art baselines.

EWISE source code is available at `https://github.com/malllabiisc/EWISE`

2 Related Work

Classical approaches to supervised WSD relied on extracting potentially relevant features and learning classifiers independently for each word

(Zhong and Ng, 2010). Extensions to use distributional word representations have been proposed (Iacobacci et al., 2016). Semi-supervised approaches learn context representations from unlabeled data, followed by a nearest neighbour classification (Melamud et al., 2016) or label propagation (Yuan et al., 2016). Recently, Raganato et al. (2017b) introduced neural sequence models for joint disambiguation of words in a sentence. All of these methods rely on sense-annotated data and, optionally, additional unlabeled corpora.

Lexical resources provide an important source of knowledge about words and their meanings. Recent work has shown that neural networks can extract semantic information from dictionary definitions (Bahdanau et al., 2017; Bosc and Vincent, 2018). In this work, we use dictionary definitions to get representations of word meanings.

Dictionary definitions have been used for WSD, motivated by the classical method of Lesk (Lesk, 1986). The original as well as subsequent modifications of the algorithm (Banerjee and Pedersen, 2003), including using word embeddings (Basile et al., 2014), operate on the hypothesis that the definition of the correct sense has a high overlap with the context in which a word is used. These methods tend to rely on heuristics based on insights about natural language text and their definitions. More recently, gloss (definition)-augmented neural approaches have been proposed which integrate a module to score definition-context similarity (Luo et al., 2018b,a), and achieve state-of-the-art results. We differ from these works in that we use the embeddings of definitions as the target space of a neural model, while learning in a supervised setup. Also, we don't rely on any overlap heuristics, and use a single definition for a given sense as provided by WordNet.

One approach for obtaining continuous representations for definitions is to use **Universal Sentence Representations**, which have been explored to allow transfer learning from large unlabeled as well as labeled data (Conneau et al., 2017; Cer et al., 2018). There has also been interest in learning deep contextualized word representations (Peters et al., 2018; Devlin et al., 2019). In this work, we evaluate definition embeddings obtained using these methods.

Structural Knowledge available in lexical resources such as WordNet has motivated several unsupervised knowledge-based approaches for WSD. Graph based techniques have been used to match words to the most relevant sense (Navigli and Lapata, 2010; Sinha and Mihalcea, 2007; Agirre et al., 2014; Moro et al., 2014; Chaplot and Salakhutdinov, 2018).

Our work differs from these methods in that we use structural knowledge to learn better representations of definitions, which are then used as targets for the WSD model. To learn a meaningful encoder for definitions we rely on knowledge graph embedding methods, where we represent an entity by the encoding of its definition. TransE (Bordes et al., 2013) models relations between entities as translations operating on the embeddings of the corresponding entities. ConvE (Dettmers et al., 2018), a more recent method, utilizes a multi-layer convolutional network, allowing it to learn more expressive features.

Predicting in an embedding space is key to our methods, allowing generalized zero shot learning capability, as well as incorporating definitions and structural knowledge. The idea has been explored in the context of zero-shot learning (Xian et al., 2018). Tying the input and output embeddings of language models (Press and Wolf, 2017) resembles our approach.

3 Background

In this work, we propose to use the training signal present in WordNet relations to learn encoders for definitions (Section 4.3.2). To learn from WordNet relations, we employ recently popular Knowledge Graph (KG) Embedding learning methods. In Section 3.1, we briefly introduce the framework for KG Embedding learning, and present the specific formulations for TransE and ConvE.

3.1 Knowledge Graph Embeddings

Knowledge Graphs, a set of relations defined over a set of entities, provide an important field of research for representation learning. Methods for learning representations for both entities and relations have been explored (Wang et al., 2017) with an aim to represent graphical knowledge. Of particular significance is the task of link prediction, i.e., predicting missing links (edges) in the graph.

A Knowledge Graph is typically comprised of a set K of N triples (h, l, t), where head h and tail t are entities, and l denotes a relation.

TransE defines a scoring function for a triple (h, l, t), as the dissimilarity between the head em-

bedding, translated by the relation embedding, and the tail embedding:

$$d_{h,l,t} = ||e_h + e_l - e_t||_2^2, \quad (1)$$

where, e_h, e_t and e_l are parameters to be learnt.

A margin based criterion, with margin γ, can then be formulated as:

$$L_T = \sum_{(h,l,t)\in K} \sum_{(h',l,t')\in K'} [\gamma + d_{h,l,t} - d_{h',l,t'}]_+, \quad (2)$$

where K' is a set of corrupted triples (Bordes et al., 2013), and $[x]_+$ refers to the positive part of x.

ConvE formulates the scoring function $\psi_l(e_h, e_t)$ for a triple (h, l, t) as:

$$\psi_l(e_h, e_t) = f(\text{vec}(f([\overline{e_h}; \overline{e_l}] * w))W)e_t, \quad (3)$$

where e_h and e_t are entity parameters, e_l is a relation parameter, $\overline{x}$ denotes a 2D reshaping of x, w denotes the filters for 2D convolution, $\text{vec}(x)$ denotes the vectorization of x, W represents a linear transformation, and f denotes a rectified linear unit.

For a given head entity h, the score $\psi_l(e_h, e_t)$ is computed with each entity in the graph as a tail. Probability estimates for the validity of a triple are obtained by applying a logistic sigmoid function to the scores:

$$p = \sigma(\psi_l(e_h, e_t)). \quad (4)$$

The model is then trained using a binary cross entropy loss:

$$L_C = -\frac{1}{N}\sum_i (t_i.log(p_i) + (1 - t_i).log(1 - p_i)), \quad (5)$$

where t_i is 1 when $(h, l, t) \in K$ and 0, otherwise.

4 EWISE

EWISE is a general WSD framework for learning from sense-annotated data, dictionary definitions and lexical knowledge bases (Figure 1).

EWISE addresses a key issue with existing supervised WSD systems. Existing systems use discrete sense labels as targets for WSD. This limits the generalization capability to only the set of annotated words in the corpus, with reliable learning only for the word-senses which occur with high relative frequency. In this work, we propose using continuous space embeddings of senses as targets for WSD, to overcome the aforementioned supervision bottleneck.

To ensure generalized zero-shot learning capability, it is important that the target sense embeddings be obtained independent of the WSD task learning. We use definitions of senses available in WordNet to obtain sense embeddings. Using Dictionary Definitions to obtain the representation for a sense enables us to benefit from the semantic overlap between definitions of different senses, while also providing a natural way to handle unseen senses.

In Section 4.1, we state the task of WSD formally. We then describe the components of EWISE in detail. Here, we briefly discuss the components:

- **Attentive Context Encoder**: EWISE uses a Bi-directional LSTM (BiLSTM) encoder to convert the sequence of tokens in the input sentence into context-aware embeddings. Self-attention is used to enhance the context for disambiguating the current word, followed by a projection layer to produce sense embeddings for each input token. The architecture is detailed in Section 4.2.

- **Definition Encoder**: In EWISE, definition embeddings are learnt independent of the WSD task. In Section 4.3.1, we detail the usage of pretrained sentence encoders as baseline models for encoding definitions. In Section 4.3.2, we detail our proposed method to learn an encoder for definitions using structural knowledge in WordNet.

4.1 The WSD Task

WSD is a classification problem for a word w (e.g., bank) in a context c, with class labels being the word senses (e.g., financial institution).

We consider the all-words WSD task, where all content words - nouns, verbs, adjectives, adverbs - need to be disambiguated (Raganato et al., 2017a). The set of all possible senses for a word is given by a predefined sense inventory, such as WordNet. In this work, we use sense candidates as provided in the evaluation framework of (Raganato et al., 2017a) which has been created using WordNet.

More precisely, given a variable-length sequence of words $x =< x^1 \dots x^T >$, we need to predict a sequence of word senses $y =<$

$y^1 \ldots y^T >$. Output word sense y^i comes from a predefined sense inventory S. During inference, the set of candidate senses S_w for input word w is assumed to be known a priori.

4.2 Attentive Context Encoder

In this section, we detail how EWISE encodes the context of a word to be disambiguated using BiLSTMs (Hochreiter and Schmidhuber, 1997). BiLSTMs have been shown to be successful for generating effective context dependent representations for words. Following Raganato et al. (2017b), we use a BiLSTM with a self-attention layer to obtain sense-aware context specific representations of words. The sense embedding for a word is obtained through a projection of the context embedding. We then train the model with independently trained sense embeddings (Section 4.3) as target embeddings.

Our model architecture is shown in Figure 1. The model processes a sequence of tokens $x^i, i \in [T]$ in a given sentence input by first representing each token with a real-valued vector representation, e^i, via an embedding matrix $W_e \in R^{|V|*d}$, where V is the vocabulary size and d is the size of the embeddings. The vector representations are then input to a 2 layer bidirectional LSTM encoder. Each word is represented by concatenating the forward h^i_f and backward h^i_b hidden state vectors of the second LSTM layer.

$$u^i = [h^i_f, h^i_b] \tag{6}$$

Following Vaswani et al. (2017), we use a scaled dot-product attention mechanism to get context information at each timestep t. Attention queries, keys and values are obtained using projection matrices W_q, W_k and W_v respectively, while the size of the projected key (d_k) is used to scale the dot-product between queries and values.

$$\begin{aligned} e^i_t &= \text{dot}(W_q u^i, W_k u^t); t \in [1, T] \\ a^i &= \text{softmax}(\frac{e^i}{\sqrt{d_k}}) \\ c^i &= \sum_{t \in [1,T]} a^i_t . W_v u^t \\ r^i &= [u^i, c^i] \end{aligned} \tag{7}$$

A projection layer (fully connected linear layer) maps this context-aware word representation r_i to v_i in the space of sense embeddings.

$$v^i = W_l r^i \tag{8}$$

During training, we multiply this with the sense embeddings of all senses in the inventory, to obtain a score for each output sense. A bias term is added to this score, where the bias is obtained as the dot product between the sense embedding and a learned parameter b. A softmax layer then generates probability estimates for each output sense.

$$\hat{p}^i_j = \text{softmax}(\text{dot}(v^i, \rho_j) + \text{dot}(b, \rho_j)); \quad \rho_j \in S \tag{9}$$

The cross entropy loss for annotated word x^i is given by:

$$L^i_{\text{wsd}} = -\sum_j (z^i_j \log(\hat{p}^i_j)), \tag{10}$$

where z^i is the one-hot representation of the target sense y^i in the sense inventory S. The network parameters are learnt by minimizing the average cross entropy loss over all annotated words in a batch.

During inference, for each word x^i, we select the candidate sense with the highest score.

$$\hat{y}^i = \text{argmax}_j(\text{dot}(v^i, \rho_j) + \text{dot}(b, \rho_j)); \quad \rho_j \in S_{x^i} \tag{11}$$

4.3 Definition Encoder

In this section, we detail how target sense embeddings are obtained in EWISE.

4.3.1 Pretrained Sentence Encoders

We use pretrained sentence representation models, InferSent (Conneau et al., 2017) and USE (Cer et al., 2018) to encode definitions, producing sense embeddings of sizes 4096 and 512, respectively.

We also experiment with deep context encoders, ELMO (Peters et al., 2018) and BERT (Devlin et al., 2019) to obtain embeddings for definitions. In each case, we encode a definition using the available pretrained models, producing a context embedding for each word in the definition. A fixed length representation is then obtained by averaging over the context embeddings of the words in the definition, from the final layer. This produces sense embeddings of sizes 1024 with both ELMO and BERT.

4.3.2 Knowledge Graph Embedding

WordNet contains a knowledge graph, where the entities of the graph are senses (synsets), and re-

	Dev	Test Datasets				Concatenation of All Test Datasets				
	SE7	SE2	SE3	SE13	SE15	Nouns	Verbs	Adj.	Adv.	ALL
WordNet S1	55.2	66.8	66.2	63.0	67.8	67.6	50.3	74.3	80.9	65.2
Non-neural baselines										
MFS (Using training data)	54.5	65.6	66.0	63.8	67.1	67.7	49.8	73.1	80.5	65.5
IMS+emb (2016)ˆ	62.6	72.2	70.4	65.9	71.5	71.9	56.6	75.9	84.7	70.1
Lesk_{ext}+emb (2014)*	56.7	63.0	63.7	66.2	64.6	70.0	51.1	51.7	80.6	64.2
$\text{UKB}_{\text{gloss}}$+w2w (2014)*	42.9	63.5	55.4	62.9	63.3	64.9	41.4	69.5	69.7	61.1
Babelfy (2014)	51.6	67.0	63.5	66.4	70.3	68.9	50.7	73.2	79.8	66.4
Context2Vec (2016) ˆ	61.3	71.8	69.1	65.6	71.9	71.2	57.4	75.2	82.7	69.6
WSD-TM (2018)	55.6	69.0	66.9	65.3	69.6	69.7	51.2	76.0	80.9	66.9
Neural baselines										
BiLSTM+att+LEX (2017b)	63.7	72.0	69.4	66.4	70.8	71.6	57.1	75.6	83.2	69.7
BiLSTM+att+LEX+POS (2017b)	64.8	72.0	69.1	66.9	71.5	71.5	57.5	75.0	83.8	69.9
GAS_{ext} (Linear) (2018b)*	–	72.4	70.1	67.1	72.1	71.9	58.1	76.4	84.7	70.4
GAS_{ext} (Concatenation) (2018b)*	–	72.2	70.5	67.2	72.6	72.2	57.7	76.6	**85.0**	70.6
CAN_{s} (2018a)*	–	72.2	70.2	69.1	72.2	73.5	56.5	76.6	83.3	70.9
HCAN (2018a)*	–	72.8	70.3	68.5	72.8	72.7	58.2	77.4	84.1	71.1
EWISE (ConvE)*	**67.3**	**73.8**	**71.1**	**69.4**	**74.5**	**74.0**	**60.2**	**78.0**	82.1	**71.8**

Table 1: *Comparison of F1-scores for fine-grained all-words WSD* on Senseval and SemEval datasets in the framework of Raganato et al. (2017a). The F1 scores on different POS tags (Nouns, Verbs, Adjectives, and Adverbs) are also reported. WordNet S1 and MFS provide most-frequent-sense baselines. * represents models which access definitions, while ˆ indicates models which don't access any external knowledge. EWISE (ConvE) is the proposed approach, where the ConvE method was used to generate the definition embeddings. Both the non-neural and neural supervised baselines presented here rely on a back-off mechanism, using WordNet S1 for words unseen during training. For each dataset, the highest score among existing systems with a statistically significant difference (unpaired t-test, $p < 0.05$) from EWISE is underlined. EWISE, which is capable of generalizing to unseen words and senses, doesn't use any back-off. EWISE consistently outperforms all supervised and knowledge-based systems, except for adverbs. Please see Section 6.1 for details. While the overall performance of EWISE is comparable to the neural baselines in terms of statistical significance, the value of EWISE lies in its ability to handle unseen and rare words and senses (See Section 6.3). Further, among the models compared, EWISE is the only system which is statistically significant (unpaired t-test, $p < 0.01$) with respect to the WordNet S1 baseline across all test datasets.

lations are defined over these senses. Example relations include hypernym and part_of. With each entity (sense), there is an associated text definition.

We propose to use WordNet relations as the training signal for learning definition encoders. The training set K is comprised of triples (h, l, t), where head h and tail t are senses, and l is a relation. Also, g_x denotes the definition of entity x, as provided by WordNet. The dataset contains 18 WordNet relations (Bordes et al., 2013).

The goal is to learn a sentence encoder for definitions and we select the BiLSTM-Max encoder architecture due to its recent success in sentence representation (Conneau et al., 2017). The words in the definition are encoded by a 2-layer BiLSTM to obtain context-aware embeddings for each word. A fixed length representation is then obtained by Max Pooling, i.e., selecting the maximum over each dimension. We denote this definition encoder by $q(.)$.

TransE We modify the dissimilarity measure in TransE (Equation 1) to represent both head (h) and tail (t) entities by an encoding of their definitions.

$$d_{h,l,t} = -\text{cosine}(q(h) + e_l, q(t)) \quad (12)$$

The parameters of the BiLSTM model q and the relation embeddings e_l are then learnt by minimizing the loss function in Equation 2.

ConvE We modify the scoring function of ConvE (Equation 3), to represent a head entity by the encoding of its definition.

$$\psi_l(e_h, e_t) = f(\text{vec}(f([\overline{q(h)}; \overline{e_l}] * w))W)e_t \quad (13)$$

Note that we represent only the head entity with an encoding of its definition while the tail entity t is still represented by parameter e_t. This helps restrict the size of the computation graph.

The parameters of the model q, e_l and e_t are then learnt by minimizing the binary cross-entropy loss function in Equation 5.

5 Experimental Setup

In this section, we provide details on the training and evaluation datasets. The training details are

captured in Appendix A.

5.1 Data

We use the English all-words WSD benchmarks for evaluating our models:

1. SensEval-2 (Palmer et al., 2001)
2. SensEval-3 (Snyder and Palmer, 2004)
3. SemEval-2013 (Navigli et al., 2013)
4. SemEval-2015 (Moro and Navigli, 2015)
5. ALL (Raganato et al., 2017a)

Following (Raganato et al., 2017b), we use SemEval-2007 (Pradhan et al., 2007) as our development set. We use SemCor 3.0 (Miller et al., 1993) as our training set. To enable a fair comparison, we used the dataset versions provided by (Raganato et al., 2017a). For our experiments, we used the definitions available in WordNet 3.0.

6 Evaluation

In this section, we aim to answer the following questions:

- Q1: How does EWISE compare to state-of-the-art methods on standardized test sets? (Section 6.1)
- Q2: What is the effect of ablating key components from EWISE? (Section 6.2)
- Q3: Does EWISE generalize to rare and unseen words (Section 6.3.1) and senses (Section 6.3.2)?
- Q4: Can EWISE learn with less annotated data? (Section 6.4)

6.1 Overall Results

In this section, we report the performance of EWISE on the fine-grained all-words WSD task, using the standardized benchmarks and evaluation methodology introduced in Raganato et al. (2017a). In Table 1, we report the F1 scores for EWISE, and compare against the best reported supervised and knowledge-based methods.

WordNet S1 is a strong baseline obtained by using the most frequent sense of a word as listed in WordNet. MFS is a most-frequent-sense baseline obtained through the sense frequencies in the training corpus.

Context2Vec (Melamud et al., 2016), an unsupervised model for learning generic context embeddings, enables a strong baseline for supervised WSD while using a simplistic approach (nearest-neighbour algorithm).

IMS+emb (Iacobacci et al., 2016) takes the classical approach of extracting relevant features and learning an SVM for WSD. Lesk_{ext}+emb (Basile et al., 2014) relies on definition-context overlap heuristics. $\text{UKB}_{\text{gloss}}$w2w (Agirre et al., 2014), Babelfy (Moro et al., 2014) and WSD-TM (Chaplot and Salakhutdinov, 2018) provide unsupervised knowledge-based methods. Among neural baselines, we compare against the neural sequence modeling approach in BiLSTM+att+LEX(+POS) (Raganato et al., 2017b). GAS (Luo et al., 2018b) and HCAN (Luo et al., 2018a) are recent neural models which exploit sense definitions. EWISE consistently outperforms all supervised and knowledge-based methods, improving upon the state-of-the-art by 0.7 point in F1 on the ALL dataset. Further, EWISE improves WSD performance across all POS tags (Table 1) except adverbs.

Back-off : Traditional supervised approaches can't handle unseen words. WordNet S1 is used as a back-off strategy for words unseen during training. EWISE is capable of generalizing to unseen words and senses and doesn't use any back-off.

6.2 Ablation Study for EWISE

Ablation on ALL dataset	
EWISE (ConvE)	**71.8**
- w/o Sense embeddings (with back-off)	69.3
- w/o Sense embeddings (w/o back-off)	61.8
WordNet S1	65.2

Table 2: *Ablation study for EWISE* (ConvE) on the ALL dataset. Removal of sense embeddings (rows 2 and 3) results in significant performance degradation, establishing their importance in WSD. Please see Section 6.2 for details.

We provide an ablation study of EWISE on the ALL dataset in Table 2. To investigate the effect of using definition embeddings in EWISE, we trained a BiLSTM model without any externally obtained sense embeddings. This model can make predictions only on words seen during training, and is evaluated with or without a back-off strategy (WordNet S1) for unseen words (row 2 and 3). The results demonstrate that incorporating sense

embeddings is key to EWISE's performance. Further, the generalization capability of EWISE is illustrated by the improvement in F1 in the absence of a back-off strategy (10.0 points).

	Test Datasets				
	SE2	SE3	SE13	SE15	ALL
USE	73.0	70.6	**70.9**	73.7	71.5
InferSent	72.7	70.2	69.9	73.7	71.2
ELMO	72.5	70.7	68.6	72.6	70.8
BERT	73.0	69.7	70.0	73.7	71.2
DeConf	71.3	67.0	67.9	73.0	69.3
TransE	72.8	**71.4**	70.5	73.1	71.6
ConvE	**73.8**	71.1	69.4	**74.5**	**71.8**

Table 3: *Comparison of F1 scores with different sense embeddings as targets for EWISE.* While pre-trained embedding methods (USE, InferSent, ELMO, BERT) and DeConf provide impressive results, the KG embedding methods (TransE and ConvE) perform competitively or better by learning to encode definitions using WordNet alone. Please see Section 6.2 for details.

Next, we investigate the impact of the choice of sense embeddings used as the target for EWISE (Table 3), on the ALL dataset. We compare definition embeddings learnt using structural knowledge (TransE, ConvE; See Section 4.3.2) against definition embeddings obtained from pre-trained sentence and context encoders (USE, InferSent, ELMO, BERT; See Section 4.3.1). We also compared with off-the-shelf sense embeddings (DeConf) (Pilehvar and Collier, 2016), where definitions are not used. The results justify the choice of learning definition embeddings to represent senses.

6.3 Detailed Results

We provide detailed results for EWISE on the ALL dataset, compared against BiLSTM-A (BiLSTM+attention) baseline which is trained to predict in the discrete label space (Raganato et al., 2017b). We also compare against WordNet S1 and knowledge-based methods, Lesk$_{ext}$+emb and Babelfy, available in the evaluation framework of Raganato et al. (2017a).

6.3.1 WSD on Rare Words

In this section, we investigate a key claim of EWISE - the ability to disambiguate unseen and rare words. We evaluate WSD models based on different frequencies of annotated words in the training set in Figure 2. EWISE outperforms the supervised as well as knowledge-based baselines for rare as well as frequent words. The bar plot

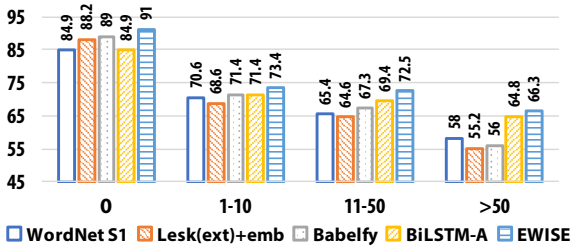


Figure 2: *Comparison of F1 scores for different frequencies of annotated words in the train set.* EWISE provides significant gains for unseen, rare as well as frequently observed annotated words. Please see Section 6.3.1 for details.

on the left (frequency=0) indicates the zero-shot learning capability of EWISE. While traditional supervised systems are limited to WordNet S1 performance (by using it as back-off for words with no annotations in the training set), EWISE provides a significant boost over both WordNet S1 as well as knowledge-based systems.

6.3.2 WSD on Rare Senses

	MFS	LFS
WordNet S1	100.0	0.0
Lesk(ext)+emb	92.7	9.4
Babelfy	93.9	12.2
BiLSTM-A	93.4	22.9
EWISE	93.5	**31.2**

Table 4: *Comparison of F1 scores on different sense frequencies.* EWISE outperforms baselines on infrequent senses, without sacrificing the performance on the most frequent sense examples. Please see Section 6.3.2 for details.

To investigate the ability to generalize to rare senses, we partition the ALL test set into two parts - the set of instances labeled with the most frequent sense of the corresponding word (MFS), and the set of remaining instances (LFS: Least Frequent Senses). Postma et al. (2016) note that existing methods learn well on the MFS set, while doing poorly ($\sim 20\%$) on the LFS set.

In Table 4, we evaluate the performance of EWISE and baseline models on MFS and LFS sets. We note that EWISE provides significant gains over a neural baseline (BiLSTM-A), as well as knowledge based methods on the LFS set, while maintaining high accuracy on the MFS set. The gain obtained on the LFS set is consistent with our hypothesis that predicting over sense embeddings enables generalization to rare senses.

6.4 Size of Training Data

	Size of training data	F1	
		Without back-off	With back-off
WordNet S1			65.2
EWISE	20%	66.8	67.0
	50%	70.1	69.2
	100%	71.8	71.0

Table 5: *Performance of EWISE with varying sizes of training data.* With only 20% of training data, EWISE is able to outperform the most-frequent-sense baseline of WordNet S1. Please see Section 6.4 for details.

In this section, we investigate if EWISE can learn efficiently from less training data, given its increased supervision bandwidth (sense embeddings instead of sense labels). In Table 5, we report the performance of EWISE on the ALL dataset with varying sizes of the training data. We note that with only 50% of training data, EWISE already competes with several supervised approaches (Table 1), while with just 20% of training data, EWISE is able to outperform the strong WordNet S1 baseline. For reference, we also present the performance of EWISE when we use back-off (WordNet S1) for words unseen during training.

7 Conclusion and Future Work

We have introduced EWISE, a general framework for learning WSD from a combination of sense-annotated data, dictionary definitions and Lexical Knowledge Bases. EWISE uses sense embeddings as targets instead of discrete sense labels. This helps the model gain zero-shot learning capabilities, demonstrated through ablation and detailed analysis. EWISE improves state-of-the-art results on standardized benchmarks for WSD. We are releasing EWISE code to promote reproducible research.

This paper should serve as a starting point to better investigate WSD on out-of-vocabulary words. Our modular architecture opens up various avenues for improvements in few-shot learning for WSD, viz., context encoder, definition encoder, and leveraging structural knowledge. Another potential future work would be to explore other ways of providing rich supervision from textual descriptions as targets.

Acknowledgments

We thank the anonymous reviewers for their constructive comments. This work is supported in part by the Ministry of Human Resource Development (Government of India), and by a travel grant from Microsoft Research India.

References

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

Dzmitry Bahdanau, Tom Bosc, Stanisaw Jastrzebski, Edward Grefenstette, Pascal Vincent, and Yoshua Bengio. 2017. Learning to compute word embeddings on the fly. *arXiv preprint arXiv:1706.00286*.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Ijcai*, volume 3, pages 805–810.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Tom Bosc and Pascal Vincent. 2018. Auto-encoding dictionary definitions into consistent word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1522–1532, Brussels, Belgium. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Devendra Singh Chaplot, Pushpak Bhattacharyya, and Ashwin Paranjape. 2015. Unsupervised word sense disambiguation using markov random field and dependency parser. In *AAAI*, pages 2217–2223.

Devendra Singh Chaplot and Ruslan Salakhutdinov. 2018. Knowledge-based word sense disambiguation using topic models. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Claudio Delli Bovi, Luis Espinosa-Anke, and Roberto Navigli. 2015. Knowledge base unification via sense embeddings and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 726–736, Lisbon, Portugal. Association for Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin, Germany. Association for Computational Linguistics.

Oier Lopez de Lacalle and Eneko Agirre. 2015. A methodology for word sense disambiguation at 90% based on large-scale CrowdSourcing. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 61–70, Denver, Colorado. Association for Computational Linguistics.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM.

Fuli Luo, Tianyu Liu, Zexue He, Qiaolin Xia, Zhifang Sui, and Baobao Chang. 2018a. Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1402–1411, Brussels, Belgium. Association for Computational Linguistics.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018b. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2473–2482, Melbourne, Australia. Association for Computational Linguistics.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 288–297, Denver, Colorado. Association for Computational Linguistics.

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

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA. Association for Computational Linguistics.

Roberto Navigli and Mirella Lapata. 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):678–692.

Steven Neale, Luís Gomes, Eneko Agirre, Oier Lopez de Lacalle, and António Branco. 2016. Word sense-aware machine translation: Including senses as contextual features for improved translation models. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2777–2783, Portorož, Slovenia. European Language Resources Association (ELRA).

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 21–24, Toulouse, France. Association for Computational Linguistics.

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

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

Mohammad Taher Pilehvar and Nigel Collier. 2016. De-conflated semantic representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1680–1690, Austin, Texas. Association for Computational Linguistics.

Marten Postma, Ruben Izquierdo Bevia, and Piek Vossen. 2016. More is not always better: balancing sense distributions for all-words word sense disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3496–3506, Osaka, Japan. The COLING 2016 Organizing Committee.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

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

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017a. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017b. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167, Copenhagen, Denmark. Association for Computational Linguistics.

Ganesh Ramakrishnan, Apurva Jadhav, Ashutosh Joshi, Soumen Chakrabarti, and Pushpak Bhattacharyya. 2003. Question answering via Bayesian inference on lexical relations. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, pages 1–10, Sapporo, Japan. Association for Computational Linguistics.

Ravi Sinha and Rada Mihalcea. 2007. Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In *Semantic Computing, 2007. ICSC 2007. International Conference on*, pages 363–369. IEEE.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain. Association for Computational Linguistics.

Kaveh Taghipour and Hwee Tou Ng. 2015. Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 314–323, Denver, Colorado. Association for Computational Linguistics.

Rocco Tripodi and Marcello Pelillo. 2017. A game-theoretic approach to word sense disambiguation. *Computational Linguistics*, 43(1):31–70.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions*

on Knowledge and Data Engineering, 29(12):2724–2743.

Dirk Weissenborn, Leonhard Hennig, Feiyu Xu, and Hans Uszkoreit. 2015. Multi-objective optimization for the joint disambiguation of nouns and named entities. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 596–605, Beijing, China. Association for Computational Linguistics.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. 2018. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1374–1385, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83, Uppsala, Sweden. Association for Computational Linguistics.

Zhi Zhong and Hwee Tou Ng. 2012. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–282, Jeju Island, Korea. Association for Computational Linguistics.

A Training Details

For both context and definition encoding, we used BiLSTMs of hidden size 2048. The input embeddings for the BiLSTM was initialized with GloVe[1] (Pennington et al., 2014) embeddings and kept fixed during training. We used the Adam optimizer for learning all our models.

WSD: We used an initial learning rate of 0.0001, a batch size of 32, and trained our models for a maximum of 200 epochs. For each run, we select the model with the best F1 score on the development set (SemEval-2007).

During training, we consider the entire sense inventory (the global pool of candidate senses of all words) for learning. During inference, for fair comparison with baselines, we disambiguate between candidates senses of a word as provided in WordNet.

TransE: We use training data from Bordes et al. (2013)[2]. We used an initial learning rate of 0.001, a batch size of 32, and trained for a maximum of 1000 epochs. The embedding size was fixed to 4096.

ConvE: We use the learning framework of Dettmers et al. (2018), and learned the model with an inital learning rate of 0.0001, a batch size of 128, label smoothing of 0.1, and a maximum of 500 epochs. We found that the best results were obtained by pretraining the entity and relation embedding using Equation 3 and then training the definition encoder using Equation 13 while allowing all parameters to train. The embedding size was fixed to 4096.

[1] http://nlp.stanford.edu/data/glove.840B.300d.zip

[2] https://everest.hds.utc.fr/lib/exe/fetch.php?media=en:wordnet-mlj12.tar.gz

We need to talk about standard splits

Kyle Gorman
City University of New York
kgorman@gc.cuny.edu

Steven Bedrick
Oregon Health & Science University
bedricks@ohsu.edu

Abstract

It is standard practice in speech & language technology to rank systems according to performance on a test set held out for evaluation. However, few researchers apply statistical tests to determine whether differences in performance are likely to arise by chance, and few examine the stability of system ranking across multiple training-testing splits. We conduct replication and reproduction experiments with nine part-of-speech taggers published between 2000 and 2018, each of which reports state-of-the-art performance on a widely-used "standard split". We fail to reliably reproduce some rankings using *randomly generated* splits. We suggest that randomly generated splits should be used in system comparison.

1 Introduction

Evaluation with a held-out test set is one of the few methodological practices shared across nearly all areas of speech and language processing. In this study we argue that one common instantiation of this procedure—evaluation with a *standard split*—is insufficient for system comparison, and propose an alternative based on multiple *random splits*.

Standard split evaluation can be formalized as follows. Let G be a set of ground truth data, partitioned into a training set G_{train}, a development set G_{dev} and a test (evaluation) set G_{test}. Let S be a system with arbitrary parameters and hyperparameters, and let $\mathcal{M}$ be an evaluation metric. Without loss of generality, we assume that $\mathcal{M}$ is a function with domain $G \times S$ and that higher values of $\mathcal{M}$ indicate better performance. Furthermore, we assume a supervised training scenario in which the free parameters of S are set so as to maximize $\mathcal{M}(G_{train}, S)$, optionally tuning hyperparameters so as to maximize $\mathcal{M}(G_{dev}, S)$. Then, if S_1 and S_2 are competing systems so trained, we prefer S_1 to S_2 if and only if $\mathcal{M}(G_{test}, S_1) > \mathcal{M}(G_{test}, S_2)$.

1.1 Hypothesis testing for system comparison

One major concern with this procedure is that it treats $\mathcal{M}(G_{test}, S_1)$ and $\mathcal{M}(G_{test}, S_2)$ as exact quantities when they are better seen as estimates of random variables corresponding to true system performance. In fact many widely used evaluation metrics, including accuracy and F-score, have known statistical distributions, allowing hypothesis testing to be used for system comparison.

For instance, consider the comparison of two systems S_1 and S_2 trained and tuned to maximize accuracy. The difference in test accuracy, $\hat{\delta} = \mathcal{M}(G_{test}, S_1) - \mathcal{M}(G_{test}, S_2)$, can be thought of as estimate of some latent variable δ representing the true difference in system performance. While the distribution of $\hat{\delta}$ is not obvious, the probability that there is no population-level difference in system performance (i.e., $\delta = 0$) can be computed indirectly using McNemar's test (Gillick and Cox, 1989). Let $n_{1>2}$ be the number of samples in G_{test} which S_1 correctly classifies but S_2 misclassifies, and $n_{2>1}$ be the number of samples which S_1 misclassifies but S_2 correctly classifies. When $\delta = 0$, roughly half of the disagreements should favor S_1 and the other half should favor S_2. Thus, under the null hypothesis, $n_{1>2} \sim \text{Bin}(n, .5)$ where $n = n_{1>2} + n_{2>1}$. And, the (one-sided) probability of the null hypothesis is the probability of sampling $n_{1>2}$ from this distribution. Similar methods can be used for other evaluation metrics, or a reference distribution can be estimated with bootstrap resampling (Efron, 1981).

Despite this, few recent studies make use of statistical system comparison. Dror et al. (2018) survey statistical practices in all long papers presented at the 2017 meeting of the Association for Computational Linguistics (ACL), and all articles published in the 2017 volume of the *Transactions of the ACL*. They find that the majority of these works

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2786–2791
Florence, Italy, July 28 - August 2, 2019.

do not use appropriate statistical tests for system comparison, and many others do not report which test(s) were used. We hypothesize that the lack of hypothesis testing for system comparison may lead to type I error, the error of rejecting a true null hypothesis. As it is rarely possible to perform the necessary hypothesis tests from published results, we evaluate this risk using a replication experiment.

1.2 Standard vs. random splits

Furthermore, we hypothesize that standard split methodology may be insufficient for system evaluation. While evaluations based on standard splits are an entrenched practice in many areas of natural language processing, the static nature of standard splits may lead researchers to unconsciously "overfit" to the vagaries of the training and test sets, producing poor generalization. This tendency may also be amplified by *publication bias* in the sense of Scargle (2000). The field has chosen to define "state of the art" performance as "the best performance on a standard split", and few experiments which do not report improvements on a standard split are ultimately published. This effect is likely to be particularly pronounced on highly-saturated tasks for which system performance is near ceiling, as this increases the prior probability of the null hypothesis (i.e., of no difference). We evaluate this risk using a series of reproductions.

1.3 Replication and reproduction

In this study we perform a replication and a series of reproductions. These techniques were until recently quite rare in this field, despite the inherently repeatable nature of most natural language processing experiments. Researchers attempting replications or reproductions have reported problems with availability of data (Mieskes, 2017; Wieling et al., 2018) and software (Pedersen, 2008), and various details of implementation (Fokkens et al., 2013; Reimers and Gurevych, 2017; Schluter and Varab, 2018). While we cannot completely avoid these pitfalls, we select a task—English part-of-speech tagging—for which both data and software are abundantly available. This task has two other important affordances for our purposes. First, it is *face-valid*, both in the sense that the equivalence classes defined by POS tags reflect genuine linguistic insights and that standard evaluation metrics such as token and sentence accuracy directly measure the underlying construct. Secondly, POS tagging is *useful* both in zero-shot settings (e.g., Elkahky et al., 2018; Trask et al., 2015) and as a source of features for many downstream tasks, and in both settings, tagging errors are likely to propagate. We release the underlying software under a permissive license.[1]

2 Materials & Methods

2.1 Data

The Wall St. Journal (WSJ) portion of Penn Treebank-3 (LDC99T42; Marcus et al., 1993) is commonly used to evaluate English part-of-speech taggers. In experiment 1, we also use a portion of OntoNotes 5 (LDC2013T19; Weischedel et al., 2011), a substantial subset of the Penn Treebank WSJ data re-annotated for quality assurance.

2.2 Models

We attempted to choose a set of taggers claiming state-of-the-art performance at time of publication. We first identified candidate taggers using the "State of the Art" page for part-of-speech tagging on the ACL Wiki.[2] We then selected nine taggers for which all needed software and external data was available at time of writing. These taggers are described in more detail below.

2.3 Metrics

Our primarily evaluation metric is token accuracy, the percentage of tokens which are correctly tagged with respect to the gold data. We compute 95% Wilson (1927) score confidence intervals for accuracies, and use the two-sided mid-p variant (Fagerland et al., 2013) of McNemar's test for system comparison. We also report out-of-vocabulary (OOV) accuracy—that is, token accuracy limited to tokens not present in the training data—and sentence accuracy, the percentage of sentences for which there are no tagging errors.

3 Results

Table 1 reports statistics for the standard split. The OntoNotes sample is slightly smaller as it omits sentences on financial news, most of which is highly redundant and idiosyncratic. However, the entire OntoNotes sample was tagged by a single experienced annotator, eliminating any annotator-specific biases in the Penn Treebank (e.g., Ratnaparkhi, 1997, 137f.).

[1] http://github.com/kylebgorman/SOTA-taggers
[2] http://aclweb.org/aclwiki/State_of_the_art

	# Sentences	# Tokens
	Penn Treebank	
Train.	38,219	912,344
Dev.	5,527	131,768
Test.	5,462	129,654
	OntoNotes	
Train.	28,905	703,955
Dev.	4,051	99,441
Test	4,059	98,277

Table 1: Summary statistics for the standard split.

3.1 Models

Three models—SVMTool (Giménez and Màrquez, 2004), MElt (Denis and Sagot, 2009), and Morče/COMPOST (Spoustová et al., 2009)—produced substantial compilation or runtime errors. However, we were able to perform replication with the remaining six models:

- **TnT** (Brants, 2000): a second-order (i.e., trigram) hidden Markov model with a suffix-based heuristic for unknown words, decoded with beam search
- **Collins (2002) tagger**: a linear model, features from Ratnaparkhi (1997), perceptron training with weight averaging, decoded with the Viterbi algorithm[3]
- **LAPOS** (Tsuruoka et al., 2011): a linear model, features from Tsuruoka et al. (2009) plus first-order lookahead, perceptron training with weight averaging, decoded locally
- **Stanford tagger** (Manning, 2011): a log-linear bidirectional cyclic dependency network, features from Toutanova et al. (2003) plus distributional similarity features, optimized with OWL-QN, decoded with the Viterbi algorithm
- **NLP4J** (Choi, 2016): a linear model, dynamically induced features, a hinge loss objective optimized with AdaGrad, decoded locally
- **Flair** (Akbik et al., 2018): a bidirectional long short-term memory (LSTM) conditional random fields (CRF) model, contextual string embedding features, a cross-entropy objective optimized with stochastic gradient descent, decoded globally

[3] We use an implementation by Yarmohammadi (2014).

3.2 Experiment 1: Replication

In experiment 1, we adopt the standard split established by Collins (2002): sections 00–18 are used for training, sections 19-21 for development, and sections 22-24 for testing, roughly a 80%-10%-10% split. We train and evaluate the six remaining taggers using this standard split. For each tagger, we train on the training set and evaluate on the test set. For taggers which support it, we also perform automated hyperparameter tuning on the development set. Results are shown in Table 2. We obtain exact replications for TnT and LAPOS, and for the remaining four taggers, our results are quite close to previously reported numbers. Token accuracy, OOV accuracy, and sentence accuracy give the same ranking, one consistent with published results. For Penn Treebank, McNemar's test on token accuracy is significant for all pairwise comparisons at $\alpha = .05$; for OntoNotes, one comparison is non-significant: LAPOS vs. Stanford ($p = .1366$).

3.3 Experiment 2: Reproduction

We now repeat these analyses across twenty randomly generated 80%–10%–10% splits. After Dror et al. (2017), we use the Bonferroni procedure to control *familywise error rate*, the probability of falsely rejecting at least one true null hypothesis. This is appropriate insofar as each individual trial (i.e, evaluation on a random split) has a non-trivial statistical dependence on other trials. Table 3 reports the number of random splits, out of twenty, where the McNemar test p-value is significant after the correction for familywise error rate. This provides a coarse estimate of how often the second system would be likely to significantly outperform the first system given a random partition of similar size. Most of these pairwise comparisons are stable across random trials. However, for example, Stanford tagger is not a significant improvement over LAPOS for nearly all random trials, and in some random trials—two for Penn Treebank, fourteen for OntoNotes—it is in fact worse. Recall also that the Stanford tagger was also not significantly better than LAPOS for OntoNotes in experiment 1.

Figure 1 shows token accuracies across the two experiments. The last row of the figure gives results for an *oracle ensemble* which correctly pre-

	Penn Treebank					OntoNotes
	Token			OOV	Sentence	Token
	Reported	Replicated	(95% CIs)	Replicated	Replicated	Reproduced
TnT	.9646	.9646	(.9636, .9656)	.8591	.4771	.9622
Collins	.9711	.9714	(.9704, .9723)	.8789	.5441	.9679
LAPOS	.9722	.9722	(.9713, .9731)	.8874	.5602	.9709
Stanford	.9732	.9735	(.9726, .9744)	.9060	.5710	.9714
NLP4J	.9764	.9742	(.9733, .9750)	.9148	.5756	.9742
Flair	.9785	.9774	(.9765, .9782)	.9287	.6111	.9790

Table 2: Previously reported, and replicated, accuracies for the standard split of the WSJ portion of Penn Treebank; we also provide token accuracies for a reproduction with the WSJ portion of OntoNotes.

		PTB	ON
TnT	vs. Collins	20	20
Collins	vs. LAPOS	20	7
LAPOS	vs. Stanford	1	0
Stanford	vs. NLP4J	19	20
NLP4J	vs. Flair	20	20

Table 3: The number of random trials (out of twenty) for which the second system has significantly higher token accuracy than the first after Bonferroni correction. PTB, Penn Treebank; ON, OntoNotes.

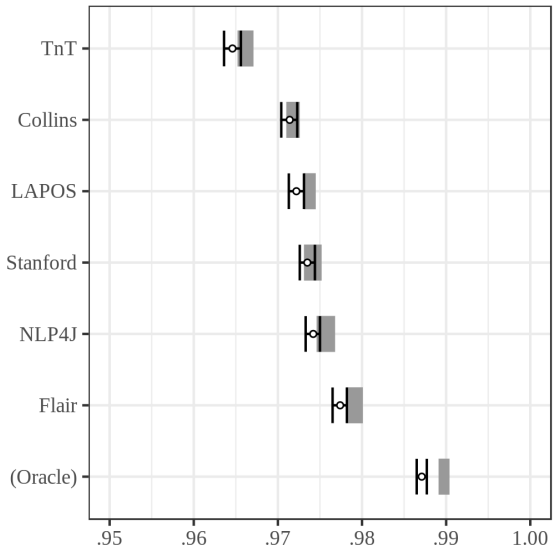


Figure 1: A visualization of Penn Treebank token accuracies in the two experiments. The whiskers shows accuracy and 95% confidence intervals in experiment 1, and shaded region represents the range of accuracies in experiment 2.

dicts the tag just in case any of the six taggers predicts the correct tag.

3.4 Error analysis

From experiment 1, we estimate that the last two decades of POS tagging research has produced a 1.28% absolute reduction in token errors. At the same time, the best tagger is 1.16% below the oracle ensemble. Thus we were interested in disagreements between taggers. We investigate this by treating each of the six taggers as separate coders in a collaborative annotation task. We compute per-sentence inter-annotator agreement using Krippendorff's α (Artstein and Poesio, 2008), then manually inspect sentences with the lowest α values, i.e., with the highest rate of disagreement. By far the most common source of disagreement are "headline"-like sentences such as *Foreign Bonds*. While these sentences are usually quite short, high disagreement is also found for some longer headlines, as in the example sentence in table 4; the effect seems to be due more to capitalization than sentence length. Several taggers lean heavily on capitalization cues to identify proper nouns, and thus capitalized tokens in headline sentences are frequently misclassified as proper nouns and vice versa, as are sentence-initial capitalized nouns in general. Most other sentences with low α have local syntactic ambiguities. For example, the word *lining*, acting as a common noun (`NN`) in the context *...a silver ____ for the...*, is mislabeled as a gerund (`VBG`) by two of six taggers.

4 Discussion

We draw attention to two distinctions between the replication and reproduction experiments. First, we find that a system judged to be significantly better than another on the basis of performance on the

	Chicken	*Chains*	*Ruffled*	*By*	*Loss*	*of*	*Customers*
Gold	NN	NNS	VBN	IN	NN	IN	NNS
TnT	NNP	NNP	NNP	IN	NN	IN	NNS
Collins	NNP	NNP	NNP	IN	NNP	IN	NNS
LAPOS	NNP	NNP	NNP	NNP	NNP	IN	NNS
Stanford	NNP	NNS	VBN	IN	NN	IN	NNS
NLP4J	NNP	NNPS	NNP	IN	NNP	IN	NNS
Flair	NN	NNS	VBN	IN	NN	IN	NNS

Table 4: Example error analysis for a Penn Treebank sentence; $\alpha = .521$.

standard split, does not in outperform that system on re-annotated data or randomly generated splits, suggesting that it is "overfit to the standard split" and does not represent a genuine improvement in performance. Secondly, as can be seen in figure 1, overall performance is slightly higher on the random splits. We posit this to be an effect of randomization at the sentence-level. For example, in the standard split the word *asbestos* occurs fifteen times in a single training set document, but just once in the test set. Such discrepancies are far less likely to arise in random splits.

Diversity of languages, data, and tasks are all highly desirable goals for natural language processing. However, nothing about this demonstration depends on any particularities of the English language, the WSJ data, or the POS tagging task. English is a somewhat challenging language for POS tagging because of its relatively impoverished inflectional morphology and pervasive noun-verb ambiguity (Elkahky et al., 2018). It would not do to use these six taggers for other languages as they are designed for English text and in some cases depend on English-only external resources for feature generation. However, random split experiments could, for instance, be performed for the subtasks of the CoNLL-2018 shared task on multilingual parsing (Zeman et al., 2018).

We finally note that repeatedly training the Flair tagger in experiment 2 required substantial grid computing resources and may not be feasible for many researchers at the present time.

5 Conclusions

We demonstrate that standard practices in system comparison, and in particular, the use of a single standard split, may result in avoidable Type I error. We suggest that practitioners who wish to firmly establish that a new system is truly state-of-the-art augment their evaluations with Bonferroni-corrected random split hypothesis testing.

It is said that statistical praxis is of greatest import in those areas of science least informed by theory. While linguistic theory and statistical learning theory both have much to contribute to part-of-speech tagging, we still lack a theory of the tagging task rich enough to guide hypothesis formation. In the meantime, we must depend on system comparison, backed by statistical best practices and error analysis, to make forward progress on this task.

Acknowledgments

We thank Mitch Marcus for valuable discussion of the Wall St. Journal data.

Steven Bedrick was supported by the National Institute on Deafness and Other Communication Disorders of the National Institutes of Health under award number R01DC015999. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

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

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Thorsten Brants. 2000. TnT: a statistical part-of-speech tagger. In *ANLC*, pages 224–231.

Jinho D. Choi. 2016. Dynamic feature induction: The last gist to the state-of-the-art. In *NAACL*, pages 271–281.

Michael Collins. 2002. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *EMNLP*, pages 1–8.

Pascal Denis and Benoît Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In *Pacific Asia Conference on Language, Information and Computation*, pages 110–119.

Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. 2017. Replicability analysis for natural language processing: testing significance with multiple datasets. *Transactions of the Association for Computational Linguistics*, 5:471–486.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *ACL*, pages 1383–1392.

Bradley Efron. 1981. Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods. *Biometrika*, 68(3):589–599.

Ali Elkahky, Kellie Webster, Daniel Andor, and Emily Pitler. 2018. A challenge set and methods for noun-verb ambiguity. In *EMNLP*, pages 2562–2572.

Morten W. Fagerland, Stian Lydersen, and Petter Laake. 2013. The McNemar test for binary matched-pairs data: mid-*p* and asymptotic are better than exact conditional. *BMC Medical Research Methodology*, 13:91–91.

Antske Fokkens, Marieke van Erp, Marten Postma, Ted Pedersen, Piek Vossen, and Nuno Freire. 2013. Offspring from reproduction problems: What replication failure teaches us. In *ACL*, pages 1691–1701.

Larry Gillick and Stephen J. Cox. 1989. Some statistical issues in the comparison of speech recognition algorithms. In *ICASSP*, pages 23–26.

Jesús Giménez and Lluís Màrquez. 2004. SVMTool: A general POS tagger generator based on support vector machines. In *LREC*, pages 43–46.

Christopher D. Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *CICLing*, pages 171–189.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Margot Mieskes. 2017. A quantitative study of data in the NLP community. In *Workshop on Ethics in NLP*, pages 23–29.

Ted Pedersen. 2008. Empiricism is not a matter of faith. *Computational Linguistics*, 34(3):465–470.

Adwait Ratnaparkhi. 1997. A maximum entropy model for part-of-speech tagging. In *EMNLP*, pages 133–142.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: performance study of LSTM-networks for sequence tagging. In *EMNLP*, pages 338–348.

Jeffrey D. Scargle. 2000. Publication bias: the "file-drawer problem" in scientific inference. *Journal of Scientific Exploration*, 14(1):91–106.

Natalie Schluter and Daniel Varab. 2018. When data permutations are pathological: the case of neural natural language inference. In *EMNLP*, pages 4935–4939.

Drahomíra Spoustová, Jan Hajič, Jan Raab, and Miroslav Spousta. 2009. Semi-supervised training for the averaged perceptron POS tagger. In *EACL*, pages 763–771.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL*, pages 173–180.

Andrew Trask, Phil Michalak, and John Liu. 2015. sense2vec: A fast and accurate method for word sense disambiguation in neural word embeddings. ArXiv preprint arXiv:1511.06388.

Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Kazama. 2011. Learning with lookahead: can history-based models rival globally optimized models? In *CoNLL*, pages 238–246.

Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. 2009. Stochastic gradient descent training for L1-regularized log-linear models with cumulative penalty. In *ICNLP-AFNLP*, pages 477–485.

Ralph Weischedel, Eduard Hovy, Mitchell P. Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, …, and Nianwen Xue. 2011. OntoNotes: a large training corpus for enhanced processing. In Joseph Olive, Caitlin Christianson, and John McCarthy, editors, *Handbook of natural language processing and machine translation*, pages 54–63. Springer, New York.

Martijn Wieling, Josine Rawee, and Gertjan van Noord. 2018. Reproducibility in computational linguistics: are we willing to share? *Computational Linguistics*, 44(4):641–649.

Edwin B. Wilson. 1927. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22:209–212.

Mahsa Yarmohammadi. 2014. Discriminative training with perceptron algorithm for POS tagging task. Technical Report CSLU-2014-001, Center for Spoken Language Understanding, Oregon Health & Science University.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, …, and Josie Li. 2018. CoNLL 2018 shared task: multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 shared task: multilingual parsing from raw text to Universal Dependencies*, pages 1–21.

A Simple Theoretical Model of Importance for Summarization

Maxime Peyrard*
EPFL
maxime.peyrard@epfl.ch

Abstract

Research on summarization has mainly been driven by empirical approaches, crafting systems to perform well on standard datasets with the notion of information *Importance* remaining latent. We argue that establishing theoretical models of *Importance* will advance our understanding of the task and help to further improve summarization systems. To this end, we propose simple but rigorous definitions of several concepts that were previously used only intuitively in summarization: *Redundancy*, *Relevance*, and *Informativeness*. Importance arises as a single quantity naturally unifying these concepts. Additionally, we provide intuitions to interpret the proposed quantities and experiments to demonstrate the potential of the framework to inform and guide subsequent works.

1 Introduction

Summarization is the process of identifying the most *important information* from a source to produce a comprehensive output for a particular user and task (Mani, 1999). While producing readable outputs is a problem shared with the field of *Natural Language Generation*, the core challenge of summarization is the identification and selection of *important information*. The task definition is rather intuitive but involves vague and undefined terms such as *Importance* and *Information*.

Since the seminal work of Luhn (1958), automatic text summarization research has focused on empirical developments, crafting summarization systems to perform well on standard datasets leaving the formal definitions of *Importance* latent (Das and Martins, 2010; Nenkova and McKeown, 2012). This view entails collecting datasets, defining evaluation metrics and iteratively selecting the best-performing systems either via supervised learning or via repeated comparison of unsupervised systems (Yao et al., 2017).

Such solely empirical approaches may lack guidance as they are often not motivated by more general theoretical frameworks. While these approaches have facilitated the development of practical solutions, they only identify signals correlating with the vague human intuition of *Importance*. For instance, structural features like centrality and repetitions are still among the most used proxies for *Importance* (Yao et al., 2017; Kedzie et al., 2018). However, such features just correlate with *Importance* in standard datasets. Unsurprisingly, simple adversarial attacks reveal their weaknesses (Zopf et al., 2016).

We postulate that theoretical models of *Importance* are beneficial to organize research and guide future empirical works. Hence, in this work, we propose a simple definition of information importance within an abstract theoretical framework. This requires the notion of information, which has received a lot of attention since the work from Shannon (1948) in the context of communication theory. Information theory provides the means to rigorously discuss the abstract concept of information, which seems particularly well suited as an entry point for a theory of summarization. However, information theory concentrates on uncertainty (entropy) about which message was chosen from a set of possible messages, ignoring the semantics of messages (Shannon, 1948). Yet, summarization is a lossy semantic compression depending on background knowledge.

In order to apply information theory to summarization, we assume texts are represented by probability distributions over so-called *semantic units* (Bao et al., 2011). This view is compatible with the common distributional embedding representation of texts rendering the presented framework applicable in practice. When applied

Research partly done at UKP Lab from TU Darmstadt.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1059–1073
Florence, Italy, July 28 - August 2, 2019.

to semantic symbols, the tools of information theory indirectly operate at the semantic level (Carnap and Bar-Hillel, 1953; Zhong, 2017).

Contributions:

- We define several concepts intuitively connected to summarization: *Redundancy*, *Relevance* and *Informativeness*. These concepts have been used extensively in previous summarization works and we discuss along the way how our framework generalizes them.
- From these definitions, we formulate properties required from a useful notion of *Importance* as the quantity unifying these concepts. We provide intuitions to interpret the proposed quantities.
- Experiments show that, even under simplifying assumptions, these quantities correlates well with human judgments making the framework promising in order to guide future empirical works.

2 Framework

2.1 Terminology and Assumptions

We call *semantic unit* an atomic piece of information (Zhong, 2017; Cruse, 1986). We note Ω the set of all possible semantic units.

A text X is considered as a semantic source emitting semantic units as envisioned by Weaver (1953) and discussed by Bao et al. (2011). Hence, we assume that X can be represented by a probability distribution $\mathbb{P}_X$ over the semantic units Ω.

Possible interpretations:
One can interpret $\mathbb{P}_X$ as the frequency distribution of semantic units in the text. Alternatively, $\mathbb{P}_X(\omega_i)$ can be seen as the (normalized) likelihood that a text X entails an atomic information ω_i (Carnap and Bar-Hillel, 1953). Another interpretation is to view $\mathbb{P}_X(\omega_i)$ as the normalized contribution (utility) of ω_i to the overall meaning of X (Zhong, 2017).

Motivation for semantic units:
In general, existing semantic information theories either postulate or imply the existence of semantic units (Carnap and Bar-Hillel, 1953; Bao et al., 2011; Zhong, 2017). For example, the *Theory of Strongly Semantic Information* produced by Floridi (2009) implies the existence of semantic units (called information units in his work). Building on this, Tsvetkov (2014) argued that the original theory of Shannon can operate at the semantic level by relying on semantic units.

In particular, existing semantic information theories imply the existence of semantic units in formal semantics (Carnap and Bar-Hillel, 1953), which treat natural languages as formal languages (Montague, 1970). In general, lexical semantics (Cruse, 1986) also postulates the existence of elementary constituents called minimal semantic constituents. For instance, with frame semantics (Fillmore, 1976), frames can act as semantic units.

Recently, distributional semantics approaches have received a lot of attention (Turian et al., 2010; Mikolov et al., 2013b). They are based on the distributional hypothesis (Harris, 1954) and the assumption that meaning can be encoded in a vector space (Turney and Pantel, 2010; Erk, 2010). These approaches also search latent and independent components that underlie the behavior of words (Gábor et al., 2017; Mikolov et al., 2013a).

While different approaches to semantics postulate different basic units and different properties for them, they have in common that *meaning arises from a set of independent and discrete units*. Thus, the semantic units assumption is general and has minimal commitment to the actual nature of semantics. This makes the framework compatible with most existing semantic representation approaches. Each approach specifies these units and can be plugged in the framework, e.g., frame semantics would define units as frames, topic models (Allahyari et al., 2017) would define units as topics and distributional representations would define units as dimensions of a vector space.

In the following paragraphs, we represent the source document(s) D and a candidate summary S by their respective distributions $\mathbb{P}_D$ and $\mathbb{P}_S$.[1]

2.2 Redundancy

Intuitively, a summary should contain a lot of information. In information-theoretic terms, the *amount of information* is measured by Shannon's

[1] We sometimes note X instead of $\mathbb{P}_X$ when it is not ambiguous

entropy. For a summary S represented by $\mathbb{P}_S$:

$$H(S) = -\sum_{\omega_i} \mathbb{P}_S(\omega_i) \cdot \log(\mathbb{P}_S(\omega_i)) \quad (1)$$

$H(S)$ is maximized for a uniform probability distribution when every semantic unit is present only once in S: $\forall(i,j), \mathbb{P}_S(\omega_i) = \mathbb{P}_S(\omega_j)$. Therefore, we define *Redundancy*, our first quantity relevant to summarization, via entropy:

$$Red(S) = H_{max} - H(S) \quad (2)$$

Since $H_{max} = \log|\Omega|$ is a constant indepedent of S, we can simply write: $Red(S) = -H(S)$.

Redundancy in Previous Works:
By definition, entropy encompasses the notion of maximum coverage. Low redundancy via maximum coverage is the main idea behind the use of submodularity (Lin and Bilmes, 2011). Submodular functions are generalizations of coverage functions which can be optimized greedily with guarantees that the result would not be far from optimal (Fujishige, 2005). Thus, they have been used extensively in summarization (Sipos et al., 2012; Yogatama et al., 2015). Otherwise, low redundancy is usually enforced during the extraction/generation procedures like MMR (Carbonell and Goldstein, 1998).

2.3 Relevance

Intuitively, observing a summary should reduce our uncertainty about the original text. A summary approximates the original source(s) and this approximation should incur a minimum loss of information. This property is usually called *Relevance*.

Here, estimating *Relevance* boils down to comparing the distributions $\mathbb{P}_S$ and $\mathbb{P}_D$, which is done via the cross-entropy $Rel(S,D) = -CE(S,D)$:

$$Rel(S,D) = \sum_{\omega_i} \mathbb{P}_S(\omega_i) \cdot \log(\mathbb{P}_D(\omega_i)) \quad (3)$$

The cross-entropy is interpreted as the average surprise of observing S while expecting D. A summary with a low expected surprise produces a low uncertainty about what were the original sources. This is achieved by exhibiting a distribution of semantic units similar to the one of the source documents: $\mathbb{P}_S \approx \mathbb{P}_D$.

Furthermore, we observe the following connection with *Redundancy*:

$$\begin{aligned} KL(S||D) &= CE(S,D) - H(S) \\ -KL(S||D) &= Rel(S,D) - Red(S) \end{aligned} \quad (4)$$

KL divergence is the information loss incurred by using D as an approximation of S (i.e., the uncertainty about D arising from observing S instead of D). A summarizer that minimizes the KL divergence minimizes *Redundancy* while maximizing *Relevance*.

In fact, this is an instance of the *Kullback Minimum Description Principle* (MDI) (Kullback and Leibler, 1951), a generalization of the *Maximum Entropy Principle* (Jaynes, 1957): the summary minimizing the KL divergence is the least biased (i.e., least redundant or with highest entropy) summary matching D. In other words, this summary fits D while inducing a minimum amount of *new* information. Indeed, any *new* information is necessarily biased since it does not arise from observations in the sources. The MDI principle and KL divergence unify *Redundancy* and *Relevance*.

Relevance in Previous Works:
Relevance is the most heavily studied aspect of summarization. In fact, by design, most unsupervised systems model *Relevance*. Usually, they used the idea of *topical frequency* where the most frequent topics from the sources must be extracted. Then, different notions of *topics* and counting heuristics have been proposed. We briefly discuss these developments here.

Luhn (1958) introduced the simple but influential idea that sentences containing the most important words are most likely to embody the original document. Later, Nenkova et al. (2006) showed experimentally that humans tend to use words appearing frequently in the sources to produce their summaries. Then, Vanderwende et al. (2007) developed the system *SumBasic*, which scores each sentence by the average probability of its words.

The same ideas can be generalized to n-grams. A prominent example is the ICSI system (Gillick and Favre, 2009) which extracts frequent bigrams. Despite being rather simple, ICSI produces strong and still close to state-of-the-art summaries (Hong et al., 2014).

Different but similar words may refer to the same topic and should not be counted separately.

This observation gave rise to a set of important techniques based on topic models (Allahyari et al., 2017). These approaches cover sentence clustering (McKeown et al., 1999; Radev et al., 2000; Zhang et al., 2015), lexical chains (Barzilay and Elhadad, 1999), Latent Semantic Analysis (Deerwester et al., 1990) or Latent Dirichlet Allocation (Blei et al., 2003) adapted to summarization (Hachey et al., 2006; Daumé III and Marcu, 2006; Wang et al., 2009; Davis et al., 2012). Approaches like hLDA can exploit repetitions both at the word and at the sentence level (Celikyilmaz and Hakkani-Tur, 2010).

Graph-based methods form another particularly powerful class of techniques to estimate the frequency of topics, e.g., via the notion of centrality (Mani and Bloedorn, 1997; Mihalcea and Tarau, 2004; Erkan and Radev, 2004). A significant body of research was dedicated to tweak and improve various components of graph-based approaches. For example, one can investigate different similarity measures (Chali and Joty, 2008). Also, different weighting schemes between sentences have been investigated (Leskovec et al., 2005; Wan and Yang, 2006).

Therefore, in existing approaches, the topics (i.e., atomic units) were words, n-grams, sentences or combinations of these. The general idea of preferring *frequent topics* based on various counting heuristics is formalized by cross-entropy. Indeed, requiring the summary to minimize the cross-entropy with the source documents implies that frequent topics in the sources should be extracted first.

An interesting line of work is based on the assumption that the best sentences are the ones that permit the best reconstruction of the input documents (He et al., 2012). It was refined by a stream of works using distributional similarities (Li et al., 2015; Liu et al., 2015; Ma et al., 2016). There, the atomic units are the dimensions of the vector spaces. This information bottleneck idea is also neatly captured by the notion of cross-entropy which is a measure of information loss. Alternatively, (Daumé and Marcu, 2002) viewed summarization as a noisy communication channel which is also rooted in information theory ideas. (Wilson and Sperber, 2008) provide a more general and less formal discussion of relevance in the context of Relevance Theory (Lavrenko, 2008).

2.4 Informativeness

Relevance still ignores other potential sources of information such as previous knowledge or preconceptions. We need to further extend the contextual boundary. Intuitively, a summary is informative if it induces, for a user, a great change in her knowledge about the world. Therefore, we introduce K, the background knowledge (or preconceptions about the task). K is represented by a probability distribution $\mathbb{P}_K$ over semantic units Ω.

Formally, the amount of *new* information contained in a summary S is given by the cross-entropy $Inf(S,K) = CE(S,K)$:

$$Inf(S,K) = -\sum_{\omega_i} \mathbb{P}_S(\omega_i) \cdot \log(\mathbb{P}_K(\omega_i)) \quad (5)$$

For *Relevance* the cross-entropy between S and D should be low. However, for *Informativeness*, the cross-entropy between S and K should be high because we measure the amount of new information induced by the summary in our knowledge.

Background knowledge is modeled by assigning a high probability to known semantic units. These probabilities correspond to the strength of ω_i in the user's memory. A simple model could be the uniform distribution over known information: $\mathbb{P}_K(\omega_i)$ is $\frac{1}{n}$ if the user knows ω_i, and 0 otherwise. However, K can control other variants of the summarization task: A personalized K_p models the preferences of a user by setting low probabilities to the semantic units of interest. Similarly, a query Q can be encoded by setting low probability to semantic units related to Q. Finally, there is a natural formulation of update summarization. Let U and D be two sets of documents. Update summarization consists in summarizing D given that the user has already seen U. This is modeled by setting $K = U$, considering U as previous knowledge.

Informativeness in Previous Works:
The modelization of *Informativeness* has received less attention by the summarization community. The problem of identifying stopwords originally faced by Luhn (1958) could be addressed by developments in the field of information retrieval using background corpora like TF·IDF (Sparck Jones, 1972). Based on the same intuition, Dunning (1993) outlined an alternative way of identifying highly descriptive words: the *log-likelihood ratio* test. Words identified with such

techniques are known to be useful in news summarization (Harabagiu and Lacatusu, 2005).

Furthermore, Conroy et al. (2006) proposed to model background knowledge by a large random set of news articles. In update summarization, Delort and Alfonseca (2012) used Bayesian topic models to ensure the extraction of informative summaries. Louis (2014) investigated background knowledge for update summarization with Bayesian surprise. This is comparable to the combination of *Informativeness* and *Redundancy* in our framework when semantic units are n-grams. Thus, previous approaches to *Informativeness* generally craft an alternate background distribution to model the *a-priori* importance of units. Then, units from the document rare in the background are preferred, which is captured by maximizing the cross-entropy between the summary and K. Indeed, unfrequent units in the background would be preferred in the summary because they would be surprising (i.e., informative) to an average user.

2.5 Importance

Since *Importance* is a measure that guides which choices to make when discarding semantic units, we must devise a way to encode their relative importance. Here, this means finding a probability distribution unifying D and K by encoding expectations about which semantic units should appear in a summary.

Informativeness requires a biased summary (w.r.t. K) and *Relevance* requires an unbiased summary (w.r.t. D). Thus, a summary should, by using only information available in D, produce what brings the most new information to a user with knowledge K. This could formalize a common intuition in summarization that units frequent in the source(s) but rare in the background are important.

Formally, let $d_i = \mathbb{P}_D(\omega_i)$ be the probability of the unit ω_i in the source D. Similarly, we note $k_i = \mathbb{P}_K(\omega_i)$. We seek a function $f(d_i, k_i)$ encoding the importance of unit ω_i. We formulate simple requirements that f should satisfy:

- Informativeness: $\forall i \neq j$, if $d_i = d_j$ and $k_i > k_j$ then $f(d_i, k_i) < f(d_j, k_j)$
- Relevance: $\forall i \neq j$, if $d_i > d_j$ and $k_i = k_j$ then $f(d_i, k_i) > f(d_j, k_j)$
- Additivity: $I(f(d_i, k_i)) \equiv \alpha I(d_i) + \beta I(k_i)$ (I is the information measure from Shannon's theory (Shannon, 1948))
- Normalization: $\sum_i f(d_i, k_i) = 1$

The first requirement states that, for two semantic units equally represented in the sources, we prefer the more informative one. The second requirement is an analogous statement for *Relevance*. The third requirement is a consistency constraint to preserve additivity of the information measures (Shannon, 1948). The fourth requirement ensures that f is a valid distribution.

Theorem 1. *The functions satisfying the previous requirements are of the form:*

$$\mathbb{P}_{\frac{D}{K}}(\omega_i) = \frac{1}{C} \cdot \frac{d_i^\alpha}{k_i^\beta} \tag{6}$$

$$C = \sum_i \frac{d_i^\alpha}{k_i^\beta}, \ \alpha, \beta \in \mathbb{R}^+ \tag{7}$$

C is the normalizing constant. The parameters α and β represent the strength given to *Relevance* and *Informativeness* respectively which is made clearer by equation (11). The proof is provided in appendix B.

Summary scoring function:
By construction, a candidate summary should approximate $\mathbb{P}_{\frac{D}{K}}$, which encodes the relative importance of semantic units. Furthermore, the summary should be non-redundant (i.e., high entropy). These two requirements are unified by the Kullback MDI principle: The least biased summary S^* that best approximates the distribution $\mathbb{P}_{\frac{D}{K}}$ is the solution of:

$$S^* = \underset{S}{\operatorname{argmax}}\, \theta_I = \underset{S}{\operatorname{argmin}}\, KL(S||\mathbb{P}_{\frac{D}{K}}) \tag{8}$$

Thus, we note θ_I as the quantity that scores summaries:

$$\theta_I(S, D, K) = -KL(\mathbb{P}_S, ||\mathbb{P}_{\frac{D}{K}}) \tag{9}$$

Interpretation of $\mathbb{P}_{\frac{D}{K}}$:
$\mathbb{P}_{\frac{D}{K}}$ can be viewed as an *importance-encoding distribution* because it encodes the relative importance of semantic units and gives an overall target for the summary.

For example, if a semantic unit ω_i is prominent in D ($\mathbb{P}_D(\omega_i)$ is high) and not known in K ($\mathbb{P}_D(\omega_i)$ is low), then $\mathbb{P}_{\frac{D}{K}}(\omega_i)$ is very high,

which means very desired in the summary. Indeed, choosing this unit will fill the gap in the knowledge K while matching the sources.

Figure 1 illustrates how this distribution behaves with respect to D and K (for $\alpha = \beta = 1$).

Summarizability:
The target distribution $\mathbb{P}_{\frac{D}{K}}$ may exhibit different properties. For example, it might be clear which semantic units should be extracted (i.e., a spiky probability distribution) or it might be unclear (i.e., many units have more or less the same importance score). This can be quantified by the entropy of the importance-encoding distribution:

$$H_{\frac{D}{K}} = H(\mathbb{P}_{\frac{D}{K}}) \quad (10)$$

Intuitively, this measures the number of possibly good summaries. If $H_{\frac{D}{K}}$ is low then $\mathbb{P}_{\frac{D}{S}}$ is spiky and there is little uncertainty about which semantic units to extract (few possible *good* summaries). Conversely, if the entropy is high, many equivalently *good* summaries are possible.

Interpretation of θ_I:
To better understand θ_I, we remark that it can be expressed in terms of the previously defined quantities:

$$\theta_I(S, D, K) \equiv -Red(S) + \alpha Rel(S, D) \quad (11)$$
$$+ \beta Inf(S, K) \quad (12)$$

Equality holds up to a constant term $\log C$ independent from S. Maximizing θ_I is equivalent to maximizing *Relevance* and *Informativeness* while minimizing *Redundancy*. Their relative strength are encoded by α and β.

Finally, $H(S)$, $CE(S, D)$ and $CE(S, K)$ are the three independent components of *Importance*.

It is worth noting that each previously defined quantity: Red, Rel and Inf are measured in bits (using base 2 for the logarithm). Then, θ_I is also an information measure expressed in bits. Shannon (1948) initially axiomatized that information quantities should be additive and therefore θ_I arising as the sum of other information quantities is unsurprising. Moreover, we ensured additivity with the third requirement of $\mathbb{P}_{\frac{D}{K}}$.

2.6 Potential Information

Relevance relates S and D, *Informativeness* relates S and K, but we can also connect D and K. Intuitively, we can extract a lot of new information from D only when K and D are different.

With the same argument laid out for *Informativeness*, we can define the amount of potential information as the average surprise of observing D while already knowing K. Again, this is given by the cross-entropy $PI_K(D) = CE(D, K)$:

$$PI_K(D) = -\sum_{\omega_i} \mathbb{P}_D(\omega_i) \cdot \log(\mathbb{P}_K(\omega_i)) \quad (13)$$

Previously, we stated that a summary should aim, using only information from D, to offer the maximum amount of new information with respect to K. $PI_K(D)$ can be understood as *Potential Information* or maximum *Informativeness*, the maximum amount of new information that a summary can extract from D while knowing K. A summary S cannot extract more than $PI_K(D)$ bits of information (if using only information from D).

3 Experiments

3.1 Experimental setup

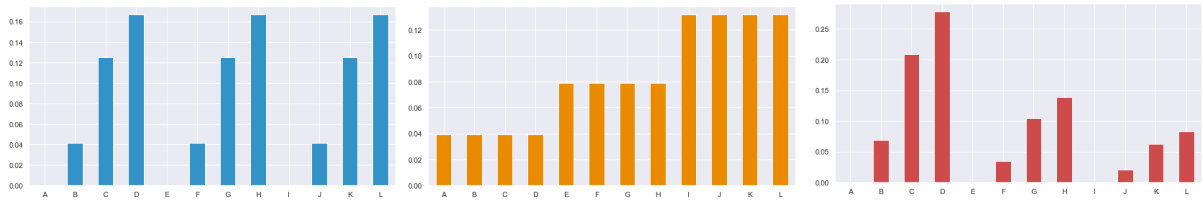
To further illustrate the workings of the formula, we provide examples of experiments done with a simplistic choice for semantic units: words. Even with simple assumptions θ_I is a meaningful quantity which correlates well with human judgments.

Data:
We experiment with standard datasets for two different summarization tasks: generic and update multi-document summarization.

We use two datasets from the Text Analysis Conference (TAC) shared task: TAC-2008 and TAC-2009.[2] In the update part, 10 new documents (B documents) are to be summarized assuming that the first 10 documents (A documents) have already been seen. The generic task consists in summarizing the initial document set (A).

For each topic, there are 4 human reference summaries and a manually created Pyramid set (Nenkova et al., 2007). In both editions, all system summaries and the 4 reference summaries were manually evaluated by NIST assessors for readability, content selection (with Pyramid) and overall responsiveness. At the time of the shared tasks, 57 systems were submitted to TAC-2008 and 55 to TAC-2009.

[2] http://tac.nist.gov/2009/Summarization/, http://tac.nist.gov/2008/



(a) ditribution $\mathbb{P}_D$ (b) distribution $\mathbb{P}_K$ (c) distribution $\mathbb{P}_{\frac{D}{K}}$

Figure 1: figure 1a represents an example distribution of sources, figure 1b an example distribution of background knowledge and figure 1c is the resulting target distribution that summaries should approximate.

Setup and Assumptions:
To keep the experiments simple and focused on illustrating the formulas, we make several simplistic assumptions. First, we choose words as semantic units and therefore texts are represented as frequency distributions over words. This assumption was already employed by previous works using information-theoretic tools for summarization (Haghighi and Vanderwende, 2009). While it is limiting, this remains a simple approximation letting us observe the quantities in action.

K, α and β are the parameters of the theory and their choice is subject to empirical investigation. Here, we make simple choices: for update summarization, K is the frequency distribution over words in the background documents (A). For generic summarization, K is the uniform probability distribution over all words from the source documents. Furthermore, we use $\alpha = \beta = 1$.

3.2 Correlation with humans

First, we measure how well the different quantities correlate with human judgments. We compute the score of each system summary according to each quantity defined in the previous section: Red, Rel, Inf, $\theta_I(S, D, K)$. We then compute the correlations between these scores and the manual Pyramid scores. Indeed, each quantity is a summary scoring function and could, therefore, be evaluated based on its ability to correlate with human judgments (Lin and Hovy, 2003). Thus, we also report the performances of the summary scoring functions from several standard baselines: **Edmundson** (Edmundson, 1969) which scores sentences based on 4 methods: term frequency, presence of cue-words, overlap with title and position of the sentence. **LexRank** (Erkan and Radev, 2004) is a popular graph-based approach which scores sentences based on their centrality in a sentence similarity graph. **ICSI** (Gillick and Favre, 2009) extracts a summary by solving a maximum coverage problem considering the most frequent bigrams in the source documents. **KL** and **JS** (Haghighi and Vanderwende, 2009) which measure the divergence between the distribution of words in the summary and in the sources. Furthermore, we report two baselines from Louis (2014) which account for background knowledge: KL_{back} and JS_{back} which measure the divergence between the distribution of the summary and the background knowledge K. Further details concerning baseline scoring functions can be found in appendix A.

We measure the correlations with Kendall's τ, a rank correlation metric which compares the orders induced by both scored lists. We report results for both generic and update summarization averaged over all topics for both datasets in table 1.

In general, the modelizations of *Relevance* (based only on the sources) correlate better with human judgments than other quantities. Metrics accounting for background knowledge work better in the update scenario. This is not surprising as the background knowledge K is more meaningful in this case (using the previous document set).

We observe that JS divergence gives slightly better results than KL. Even though KL is more theoretically appealing, JS is smoother and usually works better in practice when distributions have different supports (Louis and Nenkova, 2013).

Finally, θ_I significantly[3] outperforms all baselines in both the generic and the update case. *Red*, *Rel* and *Inf* are not particularly strong on their own, but combined together they yield a strong summary scoring function θ_I. Indeed, each quantity models only one aspect of content selection, only together they form a strong signal for *Importance*.

[3] at 0.01 with significance testing done with a t-test to compare two means

We need to be careful when interpreting these results because we made several strong assumptions: by choosing n-grams as semantic units and by choosing K rather arbitrarily. Nevertheless, these are promising results. By investigating better text representations and more realistic K, we should expect even higher correlations.

We provide a qualitative example on one topic in appendix C with a visualization of $\mathbb{P}_{\frac{D}{K}}$ in comparison to reference summaries.

	Generic	Update
ICSI	.178	.139
Edm.	.215	.205
LexRank	.201	.164
KL	.204	.176
JS	.225	.189
KL_{back}	.110	.167
JS_{back}	.066	.187
Red	.098	.096
Rel	.212	.192
Inf	.091	.086
θ_I	**.294**	**.211**

Table 1: Correlation of various information-theoretic quantities with human judgments measured by Kendall's τ on generic and update summarization.

3.3 Comparison with Reference Summaries

Intuitively, the distribution $\mathbb{P}_{\frac{D}{K}}$ should be similar to the probability distribution $\mathbb{P}_R$ of the human-written reference summaries.

To verify this, we scored the system summaries and the reference summaries with θ_I and checked whether there is a significant difference between the two lists.[4] We found that θ_I scores reference summaries significantly higher than system summaries. The $p-$value, for the generic case, is $9.2\text{e}{-6}$ and $1.1\text{e}{-3}$ for the update case. Both are much smaller than the $1\text{e}{-2}$ significance level. Therefore, θ_I is capable of distinguishing systems summaries from human written ones. For comparison, the best baseline (JS) has the following $p-$values: $8.2\text{e}{-3}$ (Generic) and $4.5\text{e}{-2}$ (Update). It does not pass the $1\text{e}{-2}$ significance level for the update scenario.

[4] with standard t-test for comparing two related means.

4 Conclusion and Future Work

In this work, we argued for the development of theoretical models of *Importance* and proposed one such framework. Thus, we investigated a theoretical formulation of the notion of *Importance*. In a framework rooted in information theory, we formalized several summary-related quantities like: *Redundancy*, *Relevance* and *Informativeness*. *Importance* arises as the notion unifying these concepts. More generally, *Importance* is the measure that guides which choices to make when information must be discarded. The introduced quantities generalize the intuitions that have previously been used in summarization research.

Conceptually, it is straightforward to build a system out of θ_I once a semantic units representation and a K have been chosen. A summarizer intends to extract or generate a summary maximizing θ_I. This fits within the general optimization framework for summarization (McDonald, 2007; Peyrard and Eckle-Kohler, 2017b; Peyrard and Gurevych, 2018)

The background knowledge and the choice of semantic units are free parameters of the theory. They are design choices which can be explored empirically by subsequent works. Our experiments already hint that strong summarizers can be developed from this framework. Characters, character n-grams, morphemes, words, n-grams, phrases, and sentences do not actually qualify as semantic units. Even though previous works who relied on information theoretic motivation (Lin et al., 2006; Haghighi and Vanderwende, 2009; Louis and Nenkova, 2013; Peyrard and Eckle-Kohler, 2016) used some of them as support for probability distributions, they are neither atomic nor independent. It is mainly because they are surface forms whereas semantic units are abstract and operate at the semantic level. However, they might serve as convenient approximations. Then, interesting research questions arise like *Which granularity offers a good approximation of semantic units? Can we automatically learn good approximations?* N-grams are known to be useful, but other granularities have rarely been considered together with information-theoretic tools.

For the background knowledge K, a promising direction would be to use the framework to actually learn it from data. In particular, one can apply supervised techniques to automatically search for K, α and β: finding the values of these parame-

ters such that θ_I has the best correlation with human judgments. By aggregating over many users and many topics one can find a generic K: what, on average, people consider as known when summarizing a document. By aggregating over different people but in one domain, one can uncover a domain-specific K. Similarly, by aggregating over many topics for one person, one would find a personalized K.

These consistute promising research directions for future works.

Acknowledgements

This work was partly supported by the German Research Foundation (DFG) as part of the Research Training Group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES) under grant No. GRK 1994/1, and via the German-Israeli Project Cooperation (DIP, grant No. GU 798/17-1). We also thank the anonymous reviewers for their comments.

References

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. 2017. Text Summarization Techniques: A Brief Survey. *International Journal of Advanced Computer Science and Applications*, 8(10).

Jie Bao, Prithwish Basu, Mike Dean, Craig Partridge, Ananthram Swami, Will Leland, and James A Hendler. 2011. Towards a theory of semantic communication. In *Network Science Workshop (NSW), 2011 IEEE*, pages 110–117. IEEE.

Regina Barzilay and Michael Elhadad. 1999. Using Lexical Chains for Text Summarization. *Advances in Automatic Text Summarization*, pages 111–121.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336.

Rudolf Carnap and Yehoshua Bar-Hillel. 1953. An Outline of a Theory of Semantic Information. *British Journal for the Philosophy of Science.*, 4.

Asli Celikyilmaz and Dilek Hakkani-Tur. 2010. A Hybrid Hierarchical Model for Multi-Document Summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 815–824, Uppsala, Sweden. Association for Computational Linguistics.

Yllias Chali and Shafiq R. Joty. 2008. Improving the performance of the random walk model for answering complex questions. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 9–12. Association for Computational Linguistics.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. 2006. Topic-Focused Multi-Document Summarization Using an Approximate Oracle Score. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 152–159, Sydney, Australia. Association for Computational Linguistics.

D.A. Cruse. 1986. *Lexical Semantics*. Cambridge University Press, Cambridge, UK.

Dipanjan Das and André F. T. Martins. 2010. A Survey on Automatic Text Summarization. *Literature Survey for the Language and Statistics II Course at CMU*.

Hal Daumé, III and Daniel Marcu. 2002. A Noisy-channel Model for Document Compression. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 449–456.

Hal Daumé III and Daniel Marcu. 2006. Bayesian Query-Focused Summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312. Association for Computational Linguistics.

Sashka T. Davis, John M. Conroy, and Judith D. Schlesinger. 2012. OCCAMS–An Optimal Combinatorial Covering Algorithm for Multi-document Summarization. In *Proceeding of the 12th International Conference on Data Mining Workshops (ICDMW)*, pages 454–463. IEEE.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Jean-Yves Delort and Enrique Alfonseca. 2012. DualSum: A Topic-model Based Approach for Update Summarization. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 214–223.

Ted Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational linguistics*, 19(1):61–74.

H. P. Edmundson. 1969. New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285.

Katrin Erk. 2010. What is Word Meaning, Really? (and How Can Distributional Models Help Us Describe It?). In *Proceedings of the 2010 workshop on geometrical models of natural language semantics*, pages 17–26. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based Lexical Centrality As Salience in Text Summarization. *Journal of Artificial Intelligence Research*, pages 457–479.

Charles J. Fillmore. 1976. Frame Semantics And the Nature of Language. *Annals of the New York Academy of Sciences*, 280(1):20–32.

Luciano Floridi. 2009. Philosophical Conceptions of Information. In *Formal Theories of Information*, pages 13–53. Springer.

Satoru Fujishige. 2005. *Submodular functions and optimization*. Annals of discrete mathematics. Elsevier, Amsterdam, Boston, Paris.

Kata Gábor, Haifa Zargayouna, Isabelle Tellier, Davide Buscaldi, and Thierry Charnois. 2017. Exploring Vector Spaces for Semantic Relations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1814–1823, Copenhagen, Denmark. Association for Computational Linguistics.

Dan Gillick and Benoit Favre. 2009. A Scalable Global Model for Summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 10–18, Boulder, Colorado. Association for Computational Linguistics.

Ben Hachey, Gabriel Murray, and David Reitter. 2006. Dimensionality Reduction Aids Term Co-Occurrence Based Multi-Document Summarization. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 1–7. Association for Computational Linguistics.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring Content Models for Multi-document Summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370.

Sanda Harabagiu and Finley Lacatusu. 2005. Topic Themes for Multi-document Summarization. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 202–209.

Zellig Harris. 1954. Distributional structure. *Word*, 10:146–162.

Zhanying He, Chun Chen, Jiajun Bu, Can Wang, Lijun Zhang, Deng Cai, and Xiaofei He. 2012. Document Summarization Based on Data Reconstruction. In *Proceeding of the Twenty-Sixth Conference on Artificial Intelligence*.

Kai Hong, John Conroy, benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A Repository of State of the Art and Competitive Baseline Summaries for Generic News Summarization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1608–1616, Reykjavik, Iceland.

Edwin T. Jaynes. 1957. Information Theory and Statistical Mechanics. *Physical Review*, 106:620–630.

Chris Kedzie, Kathleen McKeown, and Hal Daume III. 2018. Content Selection in Deep Learning Models of Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828. Association for Computational Linguistics.

Solomon Kullback and Richard A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Victor Lavrenko. 2008. *A generative theory of relevance*, volume 26. Springer Science & Business Media.

Jure Leskovec, Natasa Milic-Frayling, and Marko Grobelnik. 2005. Impact of Linguistic Analysis on the Semantic Graph Coverage and Learning of Document Extracts. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1069–1074.

Piji Li, Lidong Bing, Wai Lam, Hang Li, and Yi Liao. 2015. Reader-Aware Multi-document Summarization via Sparse Coding. In *Proceedings of the 24th International Conference on Artificial Intelligence* , pages 1270–1276.

Chin-Yew Lin, Guihong Cao, Jianfeng Gao, and Jian-Yun Nie. 2006. An Information-Theoretic Approach to Automatic Evaluation of Summaries. In *Proceedings of the Human Language Technology Conference at NAACL*, pages 463–470, New York City, USA.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1, pages 71–78.

Hui Lin and Jeff A. Bilmes. 2011. A Class of Submodular Functions for Document Summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 510–520, Portland, Oregon.

He Liu, Hongliang Yu, and Zhi-Hong Deng. 2015. Multi-document Summarization Based on Two-level Sparse Representation Model. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 196–202.

Annie Louis. 2014. A Bayesian Method to Incorporate Background Knowledge during Automatic Text Summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 333–338, Baltimore, Maryland.

Annie Louis and Ani Nenkova. 2013. Automatically Assessing Machine Summary Content Without a Gold Standard. *Computational Linguistics*, 39(2):267–300.

Hans Peter Luhn. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research Development*, 2:159–165.

Shulei Ma, Zhi-Hong Deng, and Yunlun Yang. 2016. An Unsupervised Multi-Document Summarization Framework Based on Neural Document Model. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1514–1523. The COLING 2016 Organizing Committee.

Inderjeet Mani. 1999. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA, USA.

Inderjeet Mani and Eric Bloedorn. 1997. Multi-document Summarization by Graph Search and Matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, pages 622–628, Providence, Rhode Island. AAAI Press.

Ryan McDonald. 2007. A Study of Global Inference Algorithms in Multi-document Summarization. In *Proceedings of the 29th European Conference on Information Retrieval Research*, pages 557–564.

Kathleen R. McKeown, Judith L. Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards Multidocument Summarization by Reformulation: Progress and Prospects. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, pages 453–460.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, Nevada, USA.

Richard Montague. 1970. English as a formal language. In Bruno Visentini, editor, *Linguaggi nella societa e nella tecnica*, pages 188–221. Edizioni di Communita.

Ani Nenkova and Kathleen McKeown. 2012. A Survey of Text Summarization Techniques. *Mining Text Data*, pages 43–76.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The Pyramid Method: Incorporating Human Content Selection Variation in Summarization Evaluation. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(2).

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A Compositional Context Sensitive Multi-document Summarizer: Exploring the Factors That Influence Summarization. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 573–580.

Maxime Peyrard and Judith Eckle-Kohler. 2016. A General Optimization Framework for Multi-Document Summarization Using Genetic Algorithms and Swarm Intelligence. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pages 247 – 257.

Maxime Peyrard and Judith Eckle-Kohler. 2017a. A principled framework for evaluating summarizers: Comparing models of summary quality against human judgments. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, volume Volume 2: Short Papers, pages 26–31. Association for Computational Linguistics.

Maxime Peyrard and Judith Eckle-Kohler. 2017b. Supervised learning of automatic pyramid for optimization-based multi-document summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, volume Volume 1: Long Papers, pages 1084–1094. Association for Computational Linguistics.

Maxime Peyrard and Iryna Gurevych. 2018. Objective function learning to match human judgements for optimization-based summarization. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 654–660. Association for Computational Linguistics.

Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based Summarization of Multiple Documents: Sentence Extraction, Utility-based Evaluation, and User Studies. In *Proceedings of the NAACL-ANLP Workshop on Automatic Summarization*, volume 4, pages 21–30, Seattle, Washington.

Claude E. Shannon. 1948. A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27:623–656.

Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Large-margin Learning of Submodular Summarization Models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 224–233, Avignon, France. Association for Computational Linguistics.

Karen Sparck Jones. 1972. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of documentation*, 28(1):11–21.

Victor Yakovlevich Tsvetkov. 2014. The KE Shannon and L. Floridi's Amount of Information. *Life Science Journal*, 11(11):667–671.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word Representations: A Simple and General Method for Semi-supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394.

Peter D Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of artificial intelligence research*, 37:141–188.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond SumBasic: Task-focused Summarization with Sentence Simplification and Lexical Expansion. *Information Processing & Management*, 43(6):1606–1618.

Xiaojun Wan and Jianwu Yang. 2006. Improved Affinity Graph Based Multi-Document Summarization. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 181–184. Association for Computational Linguistics.

Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. 2009. Multi-document Summarization Using Sentence-based Topic Models. In *Proceedings of the ACL-IJCNLP 2009*, pages 297–300. Association for Computational Linguistics.

Warren Weaver. 1953. Recent Contributions to the Mathematical Theory of Communication. *ETC: A Review of General Semantics*, pages 261–281.

Deirdre Wilson and Dan Sperber. 2008. *Relevance Theory*, chapter 27. John Wiley and Sons, Ltd.

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2017. Recent Advances in Document Summarization. *Knowledge and Information Systems*, 53(2):297–336.

Dani Yogatama, Fei Liu, and Noah A. Smith. 2015. Extractive Summarization by Maximizing Semantic Volume. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1961–1966, Lisbon, Portugal.

Yang Zhang, Yunqing Xia, Yi Liu, and Wenmin Wang. 2015. Clustering Sentences with Density Peaks for Multi-document Summarization. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1262–1267, Denver, Colorado. Association for Computational Linguistics.

Yixin Zhong. 2017. A Theory of Semantic Information. In *Proceedings of the IS4SI 2017 Summit Digitalisation for a Sustainable Society*, 129.

Markus Zopf, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016. Beyond Centrality and Structural Features: Learning Information Importance for Text Summarization. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016)*, pages 84–94.

A Details about Baseline Scoring Functions

In the paper, we compare the summary scoring function θ_I against the summary scoring functions derived from several summarizers following the methodology from Peyrard and Eckle-Kohler (2017a). Here, we give explicit formulation of the baseline scoring functions.

Edmundson: (Edmundson, 1969)
Edmundson (1969) presented a heuristic which scores sentences according to 4 different features:

- **Cue-phrases**: It is based on the hypothesis that the probable relevance of a sentence is affected by the presence of certain cue words such as 'significant' or 'important'. Bonus words have positive weights, stigma words have negative weights and all the others have no weight. The final score of the sentence is the sum of the weights of its words.
- **Key**: High-frequency content words are believed to be positively correlated with relevance (Luhn, 1958). Each word receives a weight based on its frequency in the document if it is not a stopword. The score of the sentence is also the sum of the weights of its words.
- **Title**: It measures the overlap between the sentence and the title.
- **Location**: It relies on the assumption that sentences appearing early or late in the source documents are more relevant.

By combining these scores with a linear combination, we can recognize the objective function:

$$\theta_{Edm.}(S) = \sum_{s \in S} \alpha_1 \cdot C(s) + \alpha_2 \cdot K(s) \quad (14)$$

$$+ \alpha_3 \cdot T(s) + \alpha_4 \cdot L(s) \quad (15)$$

The sum runs over sentences and C, K, T and L output the sentence scores for each method (Cue, Key, Title and Location).

ICSI: (Gillick and Favre, 2009)
A global linear optimization that extracts a summary by solving a maximum coverage problem of the most frequent bigrams in the source documents. ICSI has been among the best systems in a classical ROUGE evaluation (Hong et al., 2014). Here, the identification of the scoring function is trivial because it was originally formulated as an optimization task. If c_i is the i-th bigram selected in the summary and w_i is its weight computed from D, then:

$$\theta_{ICSI}(S) = \sum_{c_i \in S} c_i \cdot w_i \quad (16)$$

LexRank: (Erkan and Radev, 2004)
This is a well-known graph-based approach. A similarity graph $G(V, E)$ is constructed where V is the set of sentences and an edge e_{ij} is drawn between sentences v_i and v_j if and only if the cosine similarity between them is above a given threshold. Sentences are scored according to their PageRank score in G. Thus, $\theta_{LexRank}$ is given by:

$$\theta_{LexRank}(S) = \sum_{s \in S} PR_G(s) \quad (17)$$

Here, PR is the PageRank score of sentence s.

KL-Greedy: (Haghighi and Vanderwende, 2009)
In this approach, the summary should minimize the Kullback-Leibler (KL) divergence between the word distribution of the summary S and the word distribution of the documents D (i.e., $\theta_{KL} = -KL$):

$$\theta_{KL}(S) = -KL(S||D) \quad (18)$$

$$= -\sum_{g \in S} \mathbb{P}_S(g) \log \frac{\mathbb{P}_S(g)}{\mathbb{P}_D(g)} \quad (19)$$

$\mathbb{P}_X(w)$ represents the frequency of the word (or n-gram) w in the text X. The minus sign indicates that KL should be lower for better summaries. Indeed, we expect a good system summary to exhibit a similar probability distribution of n-grams as the sources.

Alternatively, the Jensen-Shannon (JS) divergence can be used instead of KL. Let M be the average word frequency distribution of the candidate summary S and the source documents D distribution:

$$\forall g \in S,\ \mathbb{P}_M(g) = \frac{1}{2}(\mathbb{P}_S(g) + \mathbb{P}_D(g)) \quad (20)$$

Then, the formula for JS is given by:

$$\theta_{JS}(S) = -JS(S||D) \quad (21)$$

$$= \frac{1}{2}\left(KL(S||M) + KL(D||M)\right) \quad (22)$$

Within our framework, the KL divergence acts as the unification of *Relevance* and *Redundancy* when semantic units are bigrams.

B Proof of Theorem 1

Let Ω be the set of semantic units. The notation ω_i represents one unit. Let $\mathbb{P}_T$, and $\mathbb{P}_K$ be the text representations of the source documents and background knowledge as probability distributions over semantic units.

We note $t_i = \mathbb{P}_T(\omega_i)$, the probability of the unit ω_i in the source T. Similarly, we note $k_i = \mathbb{P}_K(\omega_i)$. We seek a function f unifying T and K such that: $f(\omega_i) = f(t_i, k_i)$.

We remind the simple requirements that f should satisfy:

- Informativeness: $\forall i \neq j$, if $t_i = t_j$ and $k_i > k_j$ then $f(t_i, k_i) < f(t_j, k_j)$
- Relevance: $\forall i \neq j$, if $t_i > t_j$ and $k_i = k_j$ then $f(t_i, k_i) > f(t_j, k_j)$
- Additivity: $I(f(t_i, k_i)) \equiv \alpha I(t_i) + \beta I(k_i)$ (I is the information measure from Shannon's theory (Shannon, 1948))
- Normalization: $\sum_i f(t_i, k_i) = 1$

Theorem 1 states that the functions satisfying the previous requirements are:

$$\begin{aligned} \mathbb{P}_{\frac{T}{K}}(\omega_i) &= \frac{1}{C} \cdot \frac{t_i^\alpha}{k_i^\beta} \\ C &= \sum_i \frac{t_i^\alpha}{k_i^\beta} \text{ , } \alpha, \beta \in \mathbb{R}^+ \end{aligned} \tag{23}$$

with C the normalizing constant.

Proof. The information function defined by Shannon (1948) is the logarithm: $I = \log$. Then, the *Additivity* criterion can be written:

$$\log(f(t_i, k_i)) = \alpha \log(t_i) + \beta \log(k_i) + A \tag{24}$$

with A a constant independent of t_i and k_i

Since $\log$ is monotonous and increasing, the *Informativeness* and *Additivity* criteria can be combined:

$\forall i \neq j$, if $t_i = t_j$ and $k_i > k_j$ then:

$$\begin{gathered} \log f(t_i, k_i) < \log f(t_j, k_j) \\ \alpha \log(t_i) + \beta \log(k_i) < \alpha \log(t_j) + \beta \log(k_j) \\ \beta \log(k_i) < \beta \log(k_j) \end{gathered}$$

But $k_i > k_j$, therefore:

$$\beta < 0$$

For clarity, we can now use $-\beta$ with $\beta \in \mathbb{R}^+$.

Similarly, we can combine the *Relevance* and *Additivity* criteria: $\forall i \neq j$, if $t_i > t_j$ and $k_i = k_j$ then:

$$\begin{gathered} \log f(t_i, k_i) > \log f(t_j, k_j) \\ \alpha \log(t_i) + \beta \log(k_i) > \alpha \log(t_j) + \beta \log(k_j) \\ \alpha \log(t_i) > \alpha \log(t_j) \end{gathered}$$

But $t_i > t_j$, therefore:

$$\alpha > 0$$

Then, we have the following form from the *Additivity* criterion:

$$\begin{aligned} \log f(t_i, k_i) &= \alpha \log(t_i) - \beta \log(k_i) + A \\ f(t_i, k_i) &= e^A e^{[\alpha \log(t_i) - \beta \log(k_i)]} \\ f(t_i, k_i) &= e^A \frac{t_i^\alpha}{k_i^\beta} x \end{aligned}$$

Finally, the *Normalization* constraint specifies the constant e^A:

$$\begin{gathered} C = \frac{1}{e^A} \\ \text{and } C = \sum_i \frac{t_i^\alpha}{k_i^\beta} \\ \text{then: } A = -\log(\sum_i \frac{t_i^\alpha}{k_i^\beta}) \end{gathered}$$

□

C Example

As an example, for one selected topic of TAC-2008 update track, we computed the $\mathbb{P}_{\frac{D}{K}}$ and compare it to the distribution of the 4 reference summaries.

We report the two distributions together in figure 2. For visibility, only the top 50 words according to $\mathbb{P}_{\frac{D}{K}}$ are considered. However, we observe

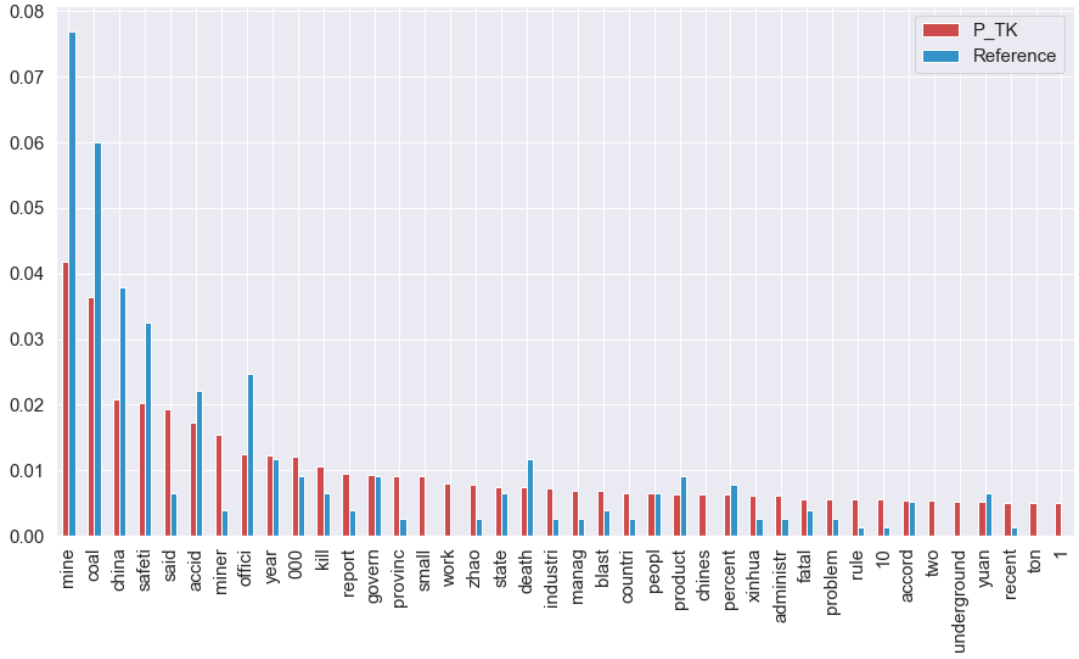


Figure 2: Example of $\mathbb{P}_{\frac{D}{K}}$ in comparison to the word distribution of reference summaries for one topic of TAC-2008 (D0803).

a good match between the distribution of the reference summaries and the *ideal* distribution as defined by $\mathbb{P}_{\frac{D}{K}}$.

Furthermore, the most desired words according to $\mathbb{P}_{\frac{D}{K}}$ make sense. This can be seen by looking at one of the human-written reference summary of this topic:

> **Reference summary for topic D0803**
> *China sacrificed coal mine safety in its massive demand for energy. Gas explosions, flooding, fires, and cave-ins cause most accidents. The mining industry is riddled with corruption from mining officials to owners. Officials are often illegally invested in mines and ignore safety procedures for production. South Africa recently provided China with information on mining safety and technology during a conference. China is beginning enforcement of safety regulations. Over 12,000 mines have been ordered to suspend operations and 4,000 others ordered closed. This year 4,228 miners were killed in 2,337 coal mine accidents. China's mines are the most dangerous worldwide.*