Generalized Supervised Attention for Text Generation*

Yixian Liu[1], Liwen Zhang[2], Xinyu Zhang[3], Yong Jiang, Yue Zhang[4], Kewei Tu[2]
[1]Interactive Entertainment Group, Tencent
[2]School of Information Science and Technology, ShanghaiTech University
[3]Language Technologies Institute, Carnegie Mellon University, USA
[4]School of Engineering, Westlake Universtiy
[4]Institute of Advanced Technology, Westlake Institute for Advanced Study
easingliu@tencent.com xinyuzh2@cs.cmu.edu
{zhanglw1, tukw}@shanghaitech.edu.cn
zhangyue@westlake.edu.cn

Abstract

The attention-based encoder-decoder framework is widely used in many natural language generation tasks. The attention mechanism builds alignments between target words and source items that facilitate text generation. Previous work proposes supervised attention that uses human knowledge to guide the attention mechanism to learn better alignments. However, well-designed supervision built from ideal alignments can be costly or even infeasible. In this paper, we build a Generalized Supervised Attention method (GSA) based on quasi alignments, which specify candidate sets of alignments and are much easier to obtain than ideal alignments. We design a Summation Cross-Entropy (SCE) loss and a Supervised Multiple Attention (SMA) structure to accommodate quasi alignments. Experiments on three text generation tasks demonstrate that GSA improves generation performance and is robust against errors in attention supervision.

1 Introduction

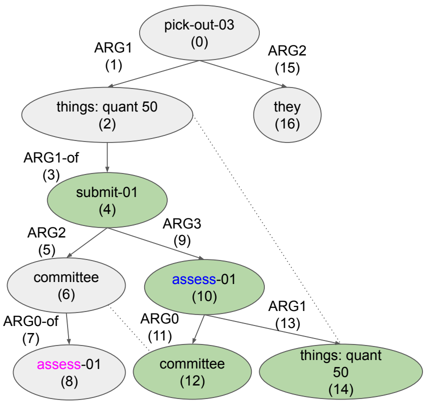
The encoder-decoder framework has been applied to various natural language generation (NLG) tasks, such as neural machine translation (Cho et al., 2014; Luong et al., 2015; Vaswani et al., 2017; Liu et al., 2016), text generation (Wiseman et al., 2017; Puduppully et al., 2019), text summarization (Liu and Lapata, 2019; Lin et al., 2018), image captioning (Anderson et al., 2018), dialogue systems (Liu et al., 2018), and so on. The attention mechanism (Bahdanau et al., 2015) plays a significant role in the framework, which automatically extracts the alignments between the target and the source for predicting the next target output. One disadvantage of the vanilla attention mechanisms is that the automatic weights do not necessarily encode prior knowledge, such as the alignments between input and output (Jain and Wallace, 2019). To alleviate this problem, supervised attention was considered (Liu et al., 2016; Mi et al., 2016; Kamigaito et al., 2017; Nguyen et al., 2018; Nguyen and Nguyen, 2018), which shows that human knowledge is helpful for guiding the learning process of attention models.

Previous work on supervised attention assumes access to ideal alignments. Unfortunately, obtaining ideal alignments is infeasible or extremely costly, for most NLG tasks. For example in Figure 1, for the AMR-to-text generation task, given the AMR graph for sentence "*From among them, pick out 50 for submission to an assessment committee to assess.*", the ideal alignment of the last word "*assess*" is node (10). As the names of (8) and (10) are the same, it is not easy to pick (10) exactly. On the other hand, it is much easier to obtain a candidate set containing both (8) and (10) and be rather confident that the ideal alignment is in the set. For different tasks, both EM-based algorithms (Brown et al., 1993; Pourdamghani et al., 2014) or rule-based methods (Flanigan et al., 2014) can be used to obtain such ambiguous alignments. However, little work has discussed making use of ambiguous labels for supervised attention.

We investigate the *generalized supervised attention* (GSA), where the supervision signal aligns a target word to multiple *possible* source items (named the *quasi alignment*), although only a subset of the items are the *true* alignment targets. The multiple source items are named *candidate set* of the quasi alignment. A generalized supervised attention framework is built for various text generation tasks with alignment relationships between target words and source items. One challenge for generalized supervised attention is that the standard Cross-Entropy (CE) loss (Liu et al., 2016)

[*] The work was done when Yixian Liu and Xinyu Zhang were students of Shanghaitech University. Kewei Tu is the corresponding author.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4991–5003
August 1–6, 2021. ©2021 Association for Computational Linguistics



Target Sentence:

From among them, pick out 50 for submission to an assessment committee to assess.

Alignments of target word assess:

Ideal alignment	(10)
Quasi alignments	(8), (10)
Relevant alignments	(4), (10), (12), (14)

Figure 1: Example of an AMR graph and the alignments between the graph and the target sentence. Ideal alignment points to the most related source node to "assess". Quasi alignments point to the source nodes with the same name. Relevant alignments point to more source items with weak relation to "assess".

can be limited because it is not suitable for quasi alignments. We design a new loss function named Summation Cross-Entropy (SCE) to replace the Cross-Entropy loss given a set of quasi alignments. SCE considers multiple candidates as a whole and is more robust against spurious candidates than traditional CE.

Supervised attention and automatically learned attention can be complementary. In Figure 1, the relevant alignments (4), (10), (12), (14) are useful for predicting "*assess*", but such alignments cannot be captured by simple rules and may require human annotation in order to be used for attention supervision. It is therefore more practical to rely on automatic attention to uncover such alignments. To balance supervised attention and automatic attention, we design a Supervised Multiple Attention (SMA) module for GSA. In SMA, there are multiple attention channels with the same structure but different parameters. One of them is used for supervised attention and the others are used for pure automatic attention (named unsupervised attention below) that are not influenced by the attention supervision. SMA can be seen as an extension of multi-head attention introduced in the Transformer (Vaswani et al., 2017).

We evaluate GSA on three real-world tasks: data-to-text generation (Koncel-Kedziorski et al., 2019), AMR-to-text generation (Mager et al., 2020), and text summarization (Yan et al., 2020). The results demonstrate that our method improves the performance in general. We also examine the robustness of our method against alignment errors. Our code will be released at https://github.com/LiuYixian/Supervised_attention.

2 Related Work

Previous work (Liu et al., 2016; Mi et al., 2016; Kamigaito et al., 2017; Nguyen et al., 2018; Nguyen and Nguyen, 2018) have studied supervised attention. Their work is based on well-designed alignments. However, no work has considered ambiguous labels, which are practically more common. We study the quasi alignments as the attention supervision and design the Summation Cross-Entropy to deal with the ambiguity in quasi alignments.

Learning with ambiguous labels has been widely studied, in which the true label is not precisely annotated but in a candidate label set. In cross-lingual Part-of-Speech, annotations are derived for low resource languages from cross-language projection, which results in partial or uncertain labels. To solve this problem, Täckström et al. (2013) proposed a partially observed conditional random field (CRF) (Lafferty et al., 2001) method, Wisniewski et al. (2014) made a history-based model, and Buys and Botha (2016) proposed an HMM-based model. SCE is designed for training attention weights using ambiguous labels. Xu et al. (2020) also study learning from ambiguous labels (called *partial label learning*) in classification tasks. Their method is based on constructing similar and dissimilar pairs of samples. However, supervised attention is not a traditional classification problem. The label spaces are various in different samples, making it difficult to construct similar pairs. Thus, the method is not suitable for GSA.

3 Basic Model

3.1 Encoder-decoder Model with Attention

Encoder-decoder models, including RNN models (Cho et al., 2014; Luong et al., 2015) and the Transformer model (Vaswani et al., 2017), are used for a variety of NLP tasks. The encoder extracts information from the source data into a memory bank, and the decoder makes use of the memory bank to generate the target sentence.

Let $\mathbf{x} = \{x_1, \ldots, x_m\}$ denote a sequence of source items, and $\mathbf{y} = \{y_1, \ldots, y_n\}$ a target sentence. The encoder converts the source data $\mathbf{x}$ into a memory bank $\mathbf{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_m\}$, where each vector $\mathbf{h}_i$ represents the contextual embedding of x_i. At the t-th time step of the decoder, the model obtains the feature vector $\mathbf{s}_t$. The meaning of $\mathbf{s}_t$ varies in different decoders. For an RNN decoder, it is the hidden state of the RNN at time t. The attention mechanism computes the contextual feature $\mathbf{c}_t$ of $\mathbf{s}_t$ over $\mathbf{H}$, which is used to predict the next target word.

The objective function of generation is the negative log conditional likelihood loss:

$$\ell(\mathbf{x}, \mathbf{y}) = -\log P(\mathbf{y}|\mathbf{x}) \tag{1}$$

3.2 Supervised Attention (SA) with Cross-Entropy (CE) Loss

Supervised attention (SA) was first introduced by Liu et al. (2016) and Mi et al. (2016) for neural machine translation. They obtain attention supervision between source and target words with off-the-shelf aligners. SA is a multi-task learning approach, where the objective function is the summation of the loss of sequence generation (*generation loss*) and the disagreement between attention distribution and attention supervision (*attention loss*) as follows:

$$Ł = \ell(\mathbf{x}, \mathbf{y}) + \lambda \sum_t \Delta(\boldsymbol{\alpha}_t, \hat{\boldsymbol{\alpha}}_t) \tag{2}$$

where $\boldsymbol{\alpha}_t$ is the computed attention and $\hat{\boldsymbol{\alpha}}_t$ is the attention supervision.

$\ell(\mathbf{x}, \mathbf{y})$ is the generation loss in Eq. (1), and λ is a positive hyper-parameter that balances the two losses. $\hat{\boldsymbol{\alpha}}_t$ is the target attention distribution and Δ measures the disagreement between the attention distribution and the target distribution. Liu et al. (2016) assume that every target word is aligned to at least one source word. If a target word is aligned to k source words, the corresponding elements in $\hat{\boldsymbol{\alpha}}_t$ are $\frac{1}{k}$ and the other elements are 0. They apply the Cross-Entropy loss as the attention loss function:

$$\Delta(\boldsymbol{\alpha}_i, \hat{\boldsymbol{\alpha}}_i) = -\sum_{j=1}^{m} \hat{\alpha}_{i,j} \times \log \alpha_{i,j} \tag{3}$$

4 Generalized Supervised Attention (GSA)

The overall architecture of GSA is shown in Figure 2. We will first introduce quasi alignments, and introduce a Summation Cross-Entropy loss function and a Supervised Multiple Attention structure.

4.1 Quasi Alignments

We consider *quasi alignments* as shown in Figure 1, in which a target word is allowed to be aligned to a candidate set in the source items, although only a subset of the candidates are the *true* alignment targets. The supervision signal provided by the quasi alignments is $\bar{\boldsymbol{\alpha}}_t = \{\bar{\alpha}_{t,1}, \ldots, \bar{\alpha}_{t,m}\}$, where $\bar{\alpha}_{t,i} = 1$ if x_i and y_t should be aligned with considerable probability. If $|\bar{\boldsymbol{\alpha}}_t| = 1$ and $\hat{\boldsymbol{\alpha}}_t$ is a one-hot alignment vector, y_t is only aligned to x_i. If $|\bar{\boldsymbol{\alpha}}_t| > 1$, $\hat{\boldsymbol{\alpha}}_t$ expresses a discrete uniform distribution over the candidate set. Such candidate items usually include some irrelevant items that should not be aligned to y_t, but it is costly to pick out the correct subset from these candidates. Therefore, we retain all these candidates and expect the training process to determine the better alignment automatically. If $|\bar{\boldsymbol{\alpha}}_t| = 0$, no item is found for y_t.

In our experiments, we obtain the quasi alignments using simple rule-based methods, which differ for different tasks, as will be discussed in Section 5.

4.2 Summation Cross-Entropy (SCE) Loss

Quasi alignments form candidate sets containing the potential aligned source items but do not indicate the true ones among them. Intuitively, we want our attention loss to penalize attention probabilities outside the candidate set but allow an arbitrary attention distribution within the set. To this end, we design the SCE loss function to maximize the total of attention probabilities in the candidate set.

$$\Delta(\boldsymbol{\alpha}_t, \bar{\boldsymbol{\alpha}}_t) = \begin{cases} 0, \text{ if } \bar{\boldsymbol{\alpha}}_t = \mathbf{0} \\ -\log(\langle \boldsymbol{\alpha}_t, \bar{\boldsymbol{\alpha}}_t \rangle), \text{ else} \end{cases} \tag{4}$$

where $\langle \cdot, \cdot \rangle$ stands for the inner product. The SCE loss is the negative logarithm of the likelihood summation of all candidate items.

Theoretically, SCE loss can be derived from a generative model. Assume that one target word should be aligned to only one true source item. For the t-th target word, we define a random variable as the true aligned source item, $P(z_t = i) = \alpha_{t,i}$. Given z_t, we re-define the candidate set as

$$\bar{\boldsymbol{\alpha}}_t = \{\bar{\alpha}_{t,1}, \ldots, \bar{\alpha}_{t,m}\}, \text{ where } \bar{\alpha}_{t,z_t} = 1. \tag{5}$$

In this way, the candidate set contains z_t. Considering that z_t is a hidden variable, the likelihood

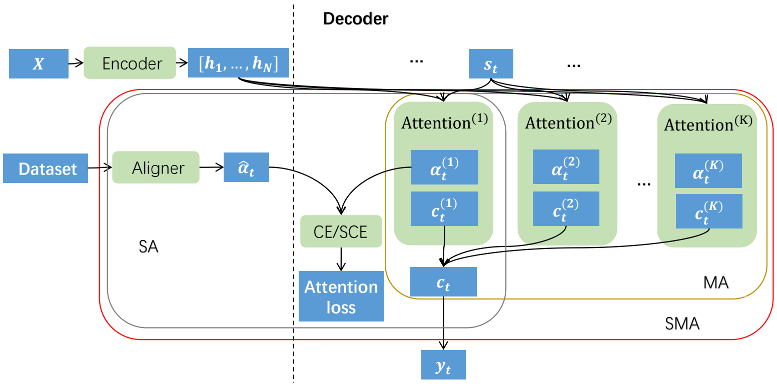


Figure 2: The overall architecture of GSA.

of the candidate set can be defined as

$$P(\bar{\boldsymbol{\alpha}}_t) = \sum_i P(z_t = i) P(\bar{\boldsymbol{\alpha}}_t | z_t = i) \quad (6)$$

$$= \sum_i \alpha_{t,i} \mathbb{I}(\bar{\alpha}_{t,i} = 1) \quad (7)$$

$$= \langle \boldsymbol{\alpha}_t, \bar{\boldsymbol{\alpha}}_t \rangle \,. \quad (8)$$

We assume that it is a certain event that we obtain a candidate set containing the true alignment given z_t. $P(\bar{\boldsymbol{\alpha}}_t | z_t)$ is a distribution over candidate sets, in which only the candidate set that contains all the words identical to x_{z_t} has probability 1 and all the other candidate sets have probability 0. Thus, $P(\bar{\boldsymbol{\alpha}}_t | z_t = i) = \mathbb{I}(\bar{\alpha}_{t,i} = 1)$. Therefore, optimizing the SCE loss is to maximize the likelihood of the candidate set.

By comparing Eq. (3) and Eq. (4), CE loss optimizes the summation of log-likelihood of target alignments, while SCE loss optimizes the log of summation of the likelihood of target alignments. If a target word is not aligned to any source items, the attention loss is 0. If it is aligned to only one source item, SCE reduces to CE (de Boer et al., 2005). If it is aligned to multiple items, then the SCE loss penalizes the attention probabilities outside the candidate set and uniformly increases the attention probabilities within the set. Note that this behavior is different from that of CE, which would encourage uniform attention over all the candidates in the set and hence produce different updates to the attention probabilities of different candidates during training.

4.3 Multiple Attention (MA) and Supervised Multiple Attention (SMA)

The motivation of supervised attention is to incorporate prior knowledge of alignments between source and target items into the attention mechanism. One problem of supervised attention is that alignments are typically established between similar items, but ideally, the decoder should also attend to some other informative source items (Figure 1), which are not necessarily similar to the target word. Besides, the automatic aligner may make errors and align the target word to irrelevant source items. Therefore, the unsupervised automatic attention mechanism is still a useful supplement to supervised attention.

We define a multiple channel attention (MA) structure, which is closely related to multi-head attention (Vaswani et al., 2017). There are K attention channels with the same structure but different parameters in MA, which work concurrently and their output contextual feature vector are combined into one contextual feature vector.

$$\mathbf{c}_t^{(k)} = \mathrm{attn}(\mathbf{s}_t, \mathbf{H}; \theta_{\mathbf{k}}) \text{ for } k = 1, \dots, K \quad (9)$$

$$\mathbf{c}_t = G(c^{(1)}, \dots, c^{(K)}), \quad (10)$$

where G is a combination function. Multi-head attention (Vaswani et al., 2017) can be regarded as a special case of the MA structure. One head of multi-head attention is an attention channel in Eq. (9). The contextual features are combined by a stacking action followed with a linear function. We do not use the standard multi-head attention because the structure of multi-head attention is strict and it is not proper for all generation tasks.

To balance supervised attention and unsupervised attention, SMA has the same structure as MA, and we compute the attention loss of the first channel only, leaving the other channels still unsupervised. The objective of SMA model is:

$$Ł = \ell(\mathbf{x}, \mathbf{y}) + \lambda \sum_t \Delta(\boldsymbol{\alpha}_t^{(1)}, \hat{\boldsymbol{\alpha}}_t) \quad (11)$$

A proposed generation model with attention can be easily modified by the SMA structure. If the original attention is one-headed, we add a new attention channel with the same structure parallel to the original one, and compute the attention loss for the new attention. The contextual features of the two attention channels are averaged in Eq. (10). If the original attention is multi-headed, we do not change the structure, and compute supervised attention loss for the first head.

5 Experiments

We apply GSA to three tasks: data-to-text generation, AMR-to-text generation, and text summarization. For each task, we choose one of the best published approaches as our basic model and modify it with GSA. In the three tasks, the relations between the source and the target are diverse. For text summarization, the source items contain more information than the target words. For data-to-text generation, the source items only contain key contents. For AMR-to-text generation, the source and the target contain the same information. We report the details of model structures and hyper-parameters in the appendix.

5.1 Data-to-text Generation

Task and Model : We consider the Abstract GENeration DAtaset (AGENDA) (Ammar et al., 2018), which contains pairs of a literature abstract and a knowledge graph extracted from the abstract. The nodes in the knowledge graphs are entity types, such as “Task” and “Method”. The edges are the relations between different entities, including “COMPARE”, “PART-OF”, and so on. We use the training, development, and test splits of 38,720/1000/1000, as Ammar et al. (2018) does.

We use GraphWriter[1] (Koncel-Kedziorski et al., 2019) on this task. The encoder of this model is a graph transformer and the decoder is an RNN decoder with attention and copying mechanism. More detail is introduced in the appendix.

Aligner: The source items of this task include entities and relations, as shown in Figure 3. We use our string matching aligner to extract the alignments from target words to the source entities and extend our aligner for the alignments of relations, such as aligning target words “*use*” and “*apply*” to source relation “USED-FOR”. For the details of the aligner for source relations, please refer to the appendix.

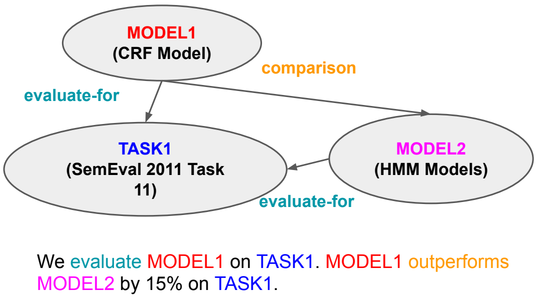


Figure 3: Example of alignments of the AGENDA dataset. The upper is part of the graph data and the lower is the corresponding sentence. The words with the same color should have quasi alignments.

Experimental Settings: We experiment with 4 different approaches in this experiment: *unsupervised attention* (UA), SA-CE, SA-SCE, SMA-SCE. The unsupervised approach means the original method without supervision on attention. As the decoder applies multi-head attention, we design the SA approach, in which the attention distributions of all heads are averaged to compute the attention loss. In this way, we consider the multi-head attention as a supervised attention channel. The SMA approach is designed as in Section 4.3, in which only the first head is a supervised attention channel. In SCE and CE approaches, we used SCE and CE loss function to supervise the attention, respectively.

5.2 AMR-to-text Generation

Task and Model : Abstract meaning representation (AMR) (Banarescu et al., 2013) is a semantic graph representation that is independent of the syntactic realization of a sentence. In the graph, nodes represent concepts and edges represent semantic relations between the concepts. AMR-to-text generation is to generate sentences from AMR graphs. We use the AMR dataset LDC2015E86, which contains 16,833 training samples, 1368 development samples, and 1371 test samples.

We use the model[2] of Mager et al. (2020) on this task, which is a GPT-2 (Radford et al., 2019) model with fine-tuning.

Aligner: We apply lemma matching to build the attention supervision as shown in Figure 1. There is a quasi alignment between a source item and target

[1] https://github.com/rikdz/GraphWriter

[2] https://github.com/IBM/GPT-too-AMR2text

Approach	Data-to-text			AMR-to-text		Summarization		
	BLEU	METEOR	ROUGE-L	BLEU	chrF++	ROUGE-1	ROUGE-2	ROUGE-L
UA	14.30	18.80	27.94	28.39	58.16	43.85	20.89	40.93
SA-CE	15.21	19.47	28.31	28.67	57.72	44.07	21.13	41.14
SA-SCE	15.49	19.80	28.62	29.03	58.44	**44.16**	**21.28**	**41.22**
SMA-SCE	**15.51**	**19.88**	**29.00**	**29.30**	**58.89**	43.9	21.09	40.91

Table 1: Main test result of GSA.

Approach	A.C.	M.C.	M.S.
Data2text	13.21%	9.41%	2
AMR2text	46.64%	27.11%	2.93
Text Sum.	83.53%	78.10%	8.98

Table 2: Alignment coverage (A.C.), multi-alignment coverage (M.C.), and average multi-alignment size (M.S.) of attention supervision.

Approach	SA-SCE	SA-CE
variance	0.1332	0.0561
entropy	0.4214	0.8364

Table 3: Normalized variance and entropy of attention probability over the candidate set.

word if they have the same lemma[3].

Experimental Settings: We experiment with 4 different approaches: UA, SA-CE, SA-SCE, SMA-SCE. We apply GSA to the multi-head attention in the last Transformer layer of the decoder.

5.3 Text Summarization

Task and Model : We use the CNN/DailyMail dataset (Nallapati et al., 2016). This dataset contains 287,226 training samples, 13,368 validation samples, and 11,490 test samples. We use ProphetNet (Yan et al., 2020) on this task, which builds a pre-training and fine-tuning method for text generation. Both the encoder and the decoder are Transformers. ProphetNet is pre-trained on a large-scale dataset (160GB).

Aligner: We obtain the quasi alignments with lemma matching as in Section 5.2. A target word is aligned to a source item if they have the same lemma. Some words, such as "*is*" and "*do*", appear very frequently and are likely to cause wrong alignments. We use the inverse document frequency (IDF) (Robertson, 2004) scores to downweight these words. More details about IDF applied here are shown in the appendix.

Experimental Settings: The model has a Transformer decoder. We set the experiments similarly to Section 5.2. The basic model is proposed by Yan et al. (2020). We cannot fully reproduce their reported result (ROUGE-1/2/L of 44.2/21.17/41.30) by running their public model[4]. Thus, we report our results.

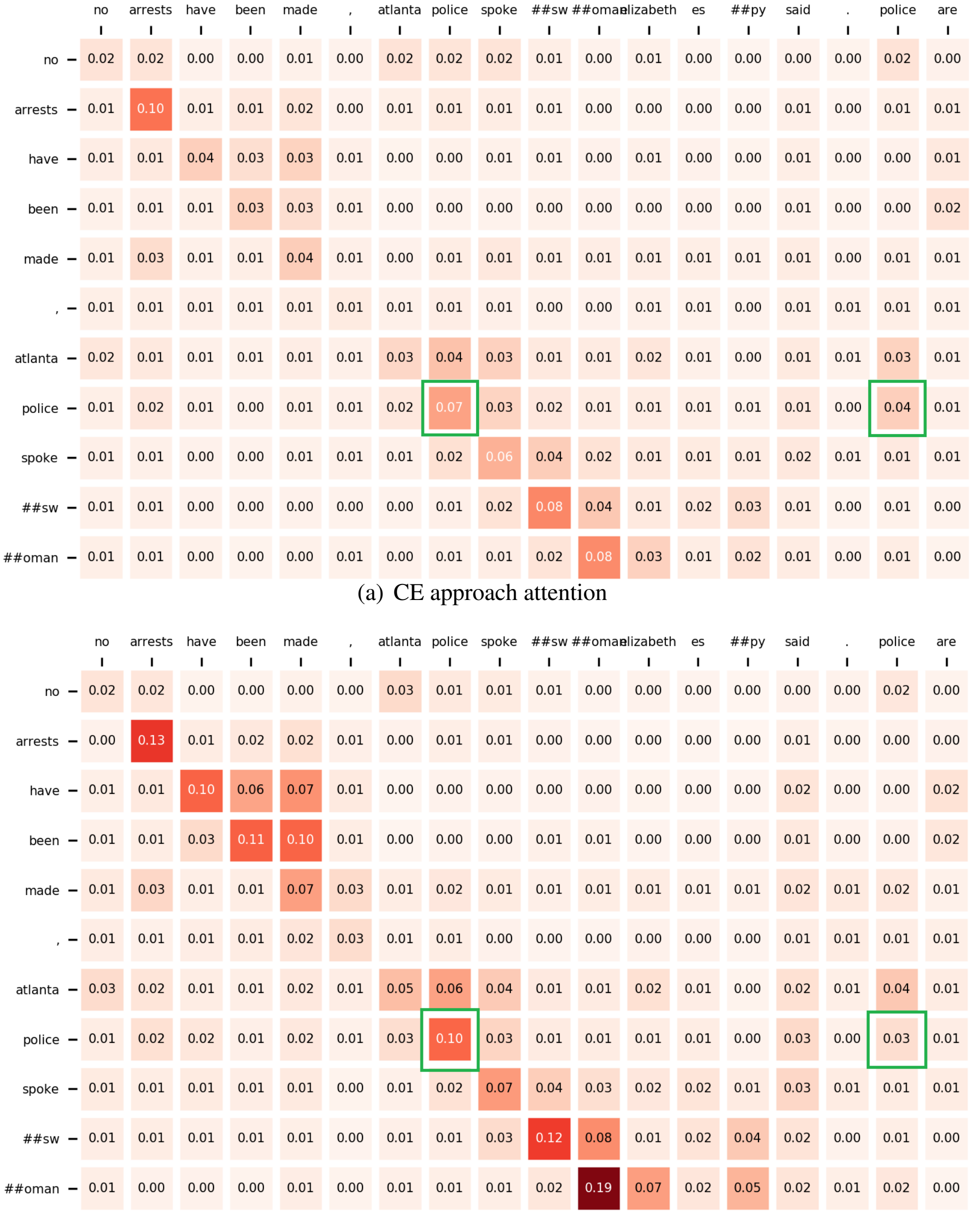
5.4 Main Results

The test results of GSA on the three tasks are shown in Table 1. For data-to-text generation, the basic model with unsupervised attention (UA) gives 14.30 BLEU, 18.80 METEOR, and 27.94 ROUGE-L. All the supervised attention approaches outperform UA. SCE outperforms CE and SMA outperforms SA. SMA-SCE achieves the best performance in BLEU, METEOR, and ROUGE-L. For AMR-to-text generation, the basic model (UA) gives a BLEU of 28.39 and a chrF++ of 58.16. All the supervised attention approaches outperform the unsupervised attention approach, demonstrating the strength of supervised attention. SMA and SCE approaches obtain higher BLEU scores compared with SA and CE, respectively. SMA-SCE achieves the best performance. For text summarization, the basic model (UA) gives a ROUGE-2 of 20.89. All the supervised attention approaches beat UA. SCE outperforms the CE. However, different from the above two tasks, the SA approaches are better than SMA. SA-SCE gives the best performance. We discuss the variation in performance of SMA compared with SA in the following.

To analyze the variations of the performance of GSA in different tasks, we compute the alignment coverage (A.C.), multi-alignment coverage (M.C.), and average multi-alignment size (M.S.) of different tasks (Table 2). A.C. is the percentage of target words with at least one alignment. M.C. is the percentage of target words with at least two alignments over all the aligned target words. M.S. is the average number of alignments of target words with at

[3] We apply Pattern (Smedt and Daelemans, 2012) to get the lemma of a word in `https://github.com/clips/pattern`.

[4] `https://github.com/microsoft/ProphetNet`.



(a) CE approach attention

(b) SCE approach attention

Figure 4: Comparison of the test attention of SCE structure.

least two alignments.

For data-to-text and AMR-to-text generation, SMA outperforms SA. On the other hand, SA performs better than SMA for text summarization. One possible reason is that the summarization dataset has much higher alignment coverage and multi-alignment coverage and the alignment accuracy may also be higher; consequently, supervised attention works so well that automatic attention becomes unnecessary or even distracting.

5.5 Significance Test

To assess the evidence of significance, we perform significance tests on GSA. The p-value is calculated using the one-tailed sign test with bootstrap resampling on the test set of all three tasks following Chollampatt et al. (2019):

- For data-to-text, we compare the Rouge-L score of SMA-SCE to the result of SA-CE.
- For AMR-to-text, we compare the BLEU score of SMA-SCE to the result of SA-CE.
- For summarization, we compare the Rouge-L score of SA-SCE to the result of SA-CE.

The p-value results are shown in Table 4, which show that the improvements are significant.

Task	Data-to-Text	Amr-to-Text	Summarization
P-value	6.5489e-12	5.5795e-10	3.925e-5

Table 4: Significance test for GSA.

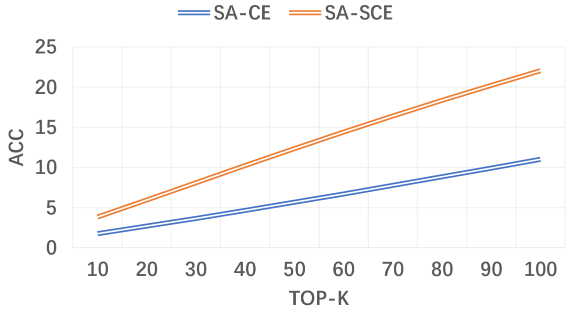


Figure 5: Top-K accuracy of SA-CE and SA-SCE in text generation.

5.6 SCE Analysis

Variance We compute the variance of attention probability in the candidate set for the text summarization task. For every generated token, we get the candidate set containing the input tokens with the same lemma as the generated token. If the candidate set contains more than one token, we compute the normalized variance and entropy of the attention scores in the candidate set. Normalized variance means that we divide every attention score by their summation and compute the variance of the normalized attention scores. Then we average the values of normalized variance and entropy in the test set. As shown in Table 3, the normalized attention variance of SCE is larger than that of CE and the entropy of SCE is smaller. It implies that CE homogenizes the attentions over the candidate set, while SCE concentrates the attentions on certain tokens. It echos Section 4.2 that CE encourages uniform attention while SCE fixes the issue.

Attention Accuracy We design an automatic evaluation method to investigate whether our SCE method can find the *correct* alignment from the quasi alignment set in an unsupervised way. For a token whose length is greater than 5[5] in the generated result, if it is matched (by lemma matching) with more than one input token, we study the generated token and the candidate set containing the matched input tokens. Specifically, we consider the local context window of length 7 around the tokens in the candidate set. The correct alignment is defined as the input token whose context window shares the most tokens with the same window around the generated token. We find the alignment selected by this automatic method almost fits the human judgment. An example is shown in Table 5.

Generated	wayne was in atlanta for a performance
Matching 1	early sunday in atlanta . no one
Matching 2	been made , atlanta police spokes woman
Matching 3	parking lot in atlanta ' s buckhead
Matching 4	wayne was in atlanta for a performance

Table 5: Automatic method to find the correct alignment.

	SA-CE	SA-SCE	SMA-SCE
BLEU	31.01	**31.19**	30.88
chrF++	59.64	**60.08**	59.86

Table 6: GSA annotated by ISI aligner for AMR-to-text generation.

The *top-K accuracy* indicates the rate that the attention score corresponding to the correct alignment is among the largest K scores. Figure 5 shows the top-K accuracy of CE and SCE for text summarization. We can find that our SA-SCE method gets higher top-K accuracy than SA-CE. That means our SCE method could find the correct alignment token and pay more attention to it without supervision.

Case Study An example of text summarization is shown in Figure 4. The figure displays a fragment of a test output sentence and the corresponding source fragment. The abscissas indicate the input text, and the ordinates denote the output summary. Figure 4(a) shows the attention of the CE approach, and 4(b) shows the attention of SCE. Both SCE and CE select the correct alignment in this example. However, the SCE approach provides higher attention probability on the correct alignment. As shown in Table 2, most output words can be flexibly aligned to more than one source word. Consider the attention probabilities of the word "*police*" framed by green squares. For one output word, there are two similar input "*police*" shown in the figure, with the first one being correct and the other one being incorrect. CE gives a probability of 0.07 for the correct alignment and 0.04 for the incorrect one. SCE approaches give the probability of 0.1 for the correct alignment and 0.03 for the incorrect one. According to section 4.2, SCE loss reduces the effect of incorrect alignments in the candidate set, which promotes the true source word.

5.7 More Powerful Supervision

In the main experiments, we apply a simple and general-purpose string matching aligner. For certain tasks, there are more powerful aligners available. To study the impact of better aligners, we in-

[5] In order to filter out high-frequency words like 'a', 'the', and 'and'.

vestigate the performance of GSA on AMR-to-text generation with the ISI aligner proposed by Pourdamghani et al. (2014), which is specially designed for the AMR-to-text task. The result is shown in Table 6. The improvement over the result in Table 1 proves that a more accurate aligner helps the supervised attention method for text generation. Besides, we can also find the result of SA-SCE better than that of SA-CE, which shows that SCE also works well while using a more accurate aligner.

We also analyze the alignments by the ISI aligner following the metrics of Table 2. The alignment coverage is 64.75%; the multi-alignment coverage is 52.17%; and the multi-alignment size is 3.11. It shows that the better aligner also produces ambiguous alignments. Therefore, SCE outperforms CE.

5.8 Robustness Analysis

We test the robustness of GSA by corrupting the attention supervision by changing correct alignments into incorrect ones. For every N target words with alignments, we change the alignments of one target word to a random and different source item. Then, we test GSA on data-to-text and AMR-to-text tasks based on the corrupted attention supervision. N ranges in $\{2, 3, 5, 10, 20\}$, which correspond to error rates of $\{50\%, 33\%, 20\%, 10\%, 5\%\}$, respectively.

The results are shown in Table 7. For AMR-to-text generation, we test the SMA-SCE approach. We observe that the supervised attention approach with a 20% error rate is still better than UA. Only with a 33% error rate does the supervised attention approach underperform the unsupervised attention. For data-to-text generation, we test the SMA-SCE approach. We observe that the SMA-SCE approach with even a 33% error rate is still better than UA. These results demonstrate the robustness of our supervised attention. On the other hand, in both experiments, the performance almost always decreases with more errors, demonstrating the importance of correct supervision.

Although GSA is shown to be robust to alignment errors, an overly high error rate would prevent the attention mechanism from finding the true alignments and make the supervised attention approaches worse than UA. Thus, reducing the mistake error rate is the most important when designing the aligner. More analyses about errors in attention supervision are in the appendix.

Error Rate	Data-to-text			AMR-to-text
	BLEU	METEOR	ROUGE-L	BLEU
0%	15.51	19.88	29.00	29.30
5%	14.93	19.68	28.44	29.01
10%	14.49	19.00	28.49	28.89
20%	14.36	18.99	28.41	28.80
33%	14.30	19.18	28.20	28.08
UA	14.30	18.80	27.94	28.39

Table 7: Robustness analysis.

6 Conclusion

We studied generalized supervised attention (GSA) for text generation tasks, considering quasi alignments instead of true alignments, which are much more difficult to obtain in practice. A Summation Cross-Entropy (SCE) loss function was designed to deal with quasi alignments, and a Supervised Multiple Attention (SMA) structure was used to balance supervised attention and unsupervised attention. Experiments on three generation tasks demonstrated that generalized supervised attention produces competitive results and is robust against errors in attention supervision.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61976139).

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew E. Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 3 (Industry Papers)*, pages 84–91.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations,* *ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, LAW-ID@ACL 2013, August 8-9, 2013, Sofia, Bulgaria*, pages 178–186.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. 2005. A tutorial on the cross-entropy method. *Annals OR*, 134(1):19–67.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguistics*, 19(2):263–311.

Jan Buys and Jan A. Botha. 2016. Cross-lingual morphological tagging for low-resource languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, 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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734.

Shamil Chollampatt, Weiqi Wang, and Hwee Tou Ng. 2019. Cross-sentence grammatical error correction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 435–445, Florence, Italy. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime G. Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1426–1436. The Association for Computer Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3543–3556. Association for Computational Linguistics.

Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. 2017. Supervised attention for sequence-to-sequence constituency parsing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017, Volume 2: Short Papers*, pages 7–12.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2284–2293.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

Junyang Lin, Xu Sun, Shuming Ma, and Qi Su. 2018. Global encoding for abstractive summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 163–169.

Lemao Liu, Masao Utiyama, Andrew M. Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 3093–3102.

Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge diffusion for neural dialogue generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1489–1498.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *CoRR*, abs/1908.08345.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421.

Manuel Mager, Ramón Fernández Astudillo, Tahira Naseem, Md. Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020. Gpt-too: A language-model-first approach for amr-to-text generation. *CoRR*, abs/2005.09123.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2283–2288.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290.

Minh Nguyen and Thien Nguyen. 2018. Who is killed by police: Introducing supervised attention for hierarchical lstms. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2277–2287.

Minh Nguyen, Toan Nguyen, and Thien Huu Nguyen. 2018. A deep learning model with hierarchical lstms and supervised attention for anti-phishing. *CoRR*, abs/1805.01554.

Nima Pourdamghani, Yang Gao, Ulf Hermjakob, and Kevin Knight. 2014. Aligning english strings with abstract meaning representation graphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 425–429.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2023–2035.

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

Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520.

Tom De Smedt and Walter Daelemans. 2012. Pattern for python. *J. Mach. Learn. Res.*, 13:2063–2067.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan T. McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Trans. Assoc. Comput. Linguistics*, 1:1–12.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2253–2263.

Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham, and François Yvon. 2014. Cross-lingual part-of-speech tagging through ambiguous learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1779–1785. ACL.

Shuang Xu, Min Yang, Yu Zhou, Ruirui Zheng, Wenpeng Liu, and Jianjun He. 2020. Partial label metric learning by collapsing classes. *INTERNATIONAL JOURNAL OF MACHINE LEARNING AND CYBERNETICS*.

Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *CoRR*, abs/2001.04063.

Appendix

A Model detail

A.1 Graph-to-text generation

In the basic model, multi-head attention is utilized to compute the contextual feature in every time step, over all the source items, including entities and relations. The copying mechanism uses basic attention just over the entities with different parameters from the multi-head attention.

The graph transformer encoder has 6 block layers. The number of attention heads is 4. The dimensions of embedding and hidden states are 500. The decoder is a one-layer LSTM recurrent networks. The decoder attention is multi-headed. In the decoding process, beam search with beam size of 4 is applied. In GSA model, the supervised attention weight is 0.5. The weight is tuned on the validation set by the BLEU score.

A.2 AMR-to-text generation

We use GPT-2 medium as the baseline generation model. It has 24 transformer block layers and 16 attention heads. The dimensions of word embeddings and hidden states are 1024. In the decoding process, the beam size is 15. The supervised attention weight for the lemma matching aligner is 0.001, and for the ISI aligner is 0.01. The weight is tuned on the validation set by the BLEU score.

A.3 Text Summarization

The baseline model is based on an encoder-decoder structure with Transformer. It has 12 block layers with 1024 hidden sizes. The beam size of decoding is 5. The supervised attention weight is 0.1. The weight is tuned on the validation set by the BLEU score.

B Aligner for Relations in Graph-to-text Generation

There are seven different relations as mentioned in the main paper. A relation can be represented by different words in the target text. For example, "use" and "apply" both suggest the relation "USED-FOR". We build a corresponding keyword list (shown in Table 8) for each relation. In the source data, each relation is of the form "a-R-b", where "a" and "b" are two entities, and "R" is the relation type. To find the alignments, we first look for the cooccurrence of "a" and "b" in the abstract with the shortest distance between them. Then, we examine the words between "a" and "b" as well as four preceding words and align the words that appear in the corresponding keyword list to both the forward and backward directions of that relation.

C IDF Score in Text Summarization

The IDF score of a word w is computed as:

$$\text{IDF}(w) = -\log \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}(w \in X^{(i)}) \qquad (12)$$

where $\mathbb{I}(\cdot)$ is the indicator function, M is the number of training samples, and $X^{(i)}$ is the target sentence of the i-th sample in the training set.

In the training step, the IDF scores of target words are used to downweight the attention loss:

$$Ł = \text{LOSS}(\mathbf{x}, \mathbf{y}) + \lambda \sum_{t} \big(\text{IDF}(w_t) \cdot \Delta(\boldsymbol{\alpha}_t, \hat{\boldsymbol{\alpha}}_t)\big), \qquad (13)$$

where w_t is the t-th word in sentence $\mathbf{y}$.

In the loss function, the attention loss of a target word is scaled by its IDF score. The IDF scores are only applied in this experiment because the alignments of these high-frequency words are rare in previous experiments.

D Alignment Error Analysis

The performance of supervised attention is influenced by the quality of the aligner. There are three types of alignment errors.

- Missing: a target word is not aligned to any source item.
- Redundancy: a target word is aligned not only to the correct source items but also to irrelevant items.
- Mistake: a target word is only aligned to some irrelevant items but not aligned to correct source items.

Missing errors reduce the alignment coverage over the target sentences. They decrease the number of target words that receive attention supervision but will not make supervised attention worse than the unsupervised baseline.

Redundancy errors, which are related to the candidate set from the flexible alignments, are handled by our SCE loss. In the worst case, a target word is aligned to all the source items, and the attention loss of this word becomes 0, resulting in no supervision. Thus, redundancy errors will not make supervised attention worse than the unsupervised baseline either. A case study is provided in the appendix showing that our method is not confused by the redundancy errors.

We empirically analyze mistake errors in section 4.5. Although supervised attention is shown to be robust to mistake errors, an overly high error rate would prevent the attention mechanism from finding the correct alignments and make the supervised attention approaches worse than the baseline. Thus, reducing the mistake error rate is the most important when designing the aligner.

Relation	Keywords
USED-FOR	“present”, “propose”, “proposes”, “proposed”, “use”, “used”, “apply”, “applied”, “application”, “applications”, “exploit”, “introduce”, “improve”, “improves”, “for”, “learned”, “obtained”, “derived”, “use”, “uses”, “using”, “based”, “exploiting”
CONJUNCTION	“addition”, “versus”
FEATURE-OF	“about”, “feature”, “stand”, “denote”
PART-OF	“incorporate”, “incorporating”, “integrate”, “integrating”, “incorporates”, “include”, “includes”, “composed”, “combines”, “combining”, “consist”, “consists”, “consisting”, “incorporate”, “incorporates”, “incorporating”, “integrate”, “integrates”, “integrating”, “contain”, “contains”, “containing”
COMPARE	“outperform”, “outperforms”, “compare”, “compared”, “more”, “than”, “outperform”, “outperforms”, “compared”
EVALUATE-FOR	“experiments”, “improvements”, “evaluated”, “improve”, “improves”
HYPONYM-OF	“such”, “including”, “namely”, “called”, “like”, “named”

Table 8: Keyword lists used in the aligner for relations in graph-to-text generation.