Reconstructing Implicit Knowledge with Language Models

Maria Becker, Siting Liang, Anette Frank
Department of Computational Linguistics, Heidelberg University
mbecker|liang|frank@cl.uni-heidelberg.de

Abstract

In this work we propose an approach for generating statements that explicate implicit knowledge connecting sentences in text. We make use of pre-trained language models which we refine by fine-tuning them on specifically prepared corpora that we enriched with implicit information, and by constraining them with relevant concepts and connecting commonsense knowledge paths. Manual and automatic evaluation of the generations shows that by refining language models as proposed, we can generate coherent and grammatically sound sentences that explicate implicit knowledge which connects sentence pairs in texts – on both in-domain and out-of-domain test data.

1 Introduction

In everyday communication and in texts people usually omit information that seems clear and evident, such that only part of the message needs to be expressed in words. In the following sentence:

(1-i) *Students should be allowed to use computers during the lectures,* (1-ii) *even though that bears the risk that they are writing emails instead of listening to the teacher.*

in order to understand the connection between (i) and (ii) we must know that *Computers are used for sending emails*, or that *Lectures are given by teachers*. Such **implicit** knowledge can easily be inferred by humans, since it is part of their background knowledge. By contrast, for computational systems implicitness in texts represents a challenge.

In this work we propose an approach for generating implicit knowledge sentences *in-between* contiguous sentences, which explicate their logical connection, utilizing pre-trained language models (LMs) that we refine as follows: i) we inject 'explanatory' knowledge by fine-tuning LMs on specifically prepared corpora, and (ii) condition text generation through constraints in form of relevant concepts and knowledge paths. Our work is inspired by the recent success of pre-trained LMs (Devlin et al., 2018; Radford et al., 2019; Yang et al., 2019a) in various downstream NLP tasks, including text generation and NL inference (Wang et al., 2018). However, for the task of *reconstructing implicit knowledge*, such LMs need to be carefully guided, not only to yield coherent statements, but to also ensure that they convey the missing, implicit information that connects given sentences in a text. To this end we create corpora with sentence pairs enriched with implicit information based on on Generics-KB (Bhakthavatsalam et al., 2020) and e-SNLI (Camburu et al., 2018), which we use for LM **fine-tuning**. For improved performance we explore methods of **constrained** language generation, guiding the model by way of relevant concepts and connecting commonsense knowledge paths.

We aim to build a system that is not limited to specific text genres or knowledge domains, and thus evaluate our models **in-domain** – on testsets from our fine-tuning corpora; and **out-of-domain** – using IKAT (Becker et al., 2020), an argumentative corpus which offers sentence pairs annotated with implicit knowledge that connects them.

A central contribution of this work is an **in-depth evaluation** of the quality of generations delivered by different model variants, and their ability of expressing implicitly conveyed knowledge. We propose a **manual** evaluation setup covering four dimensions – grammaticality, coherence, content, and comparison to gold references – , and compare these to various **automatic** evaluation metrics. Our experiments show that with our proposed approach we can generate coherent sentences that explicate implicit knowledge that connects given sentence pairs; and that current text generation metrics are not sufficient to evaluate this challenging task.

Our **contributions** are: (i) We empirically compare different types of LMs, exploring which model is best suited for the task of generating sentences that express implicit information between sen-

Proceedings of Deep Learning Inside Out (DeeLIO):
The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 11–24
Online, June 10, 2021. ©2021 Association for Computational Linguistics

tences. (ii) We create datasets that include implicit information holding between sentence pairs, which we use for fine-tuning our LMs, and which can be used for general commonsense reasoning tasks. (iii) We propose a method for constrained generation by injecting concepts or commonsense knowledge paths as language modeling constraints, and show that key concepts, and even more, knowledge paths improve the quality of generations. (iv) We carefully evaluate the quality of the generated implicit knowledge sentences, both manually and automatically, and discuss strengths and limitations of automatic similarity metrics.[1]

2 Related Work

Recent progress in **pretraining LMs** on large text corpora led to improvements for various downstream NLP tasks. It has also been shown that knowledge acquired during pre-training can be leveraged by **fine-tuning** these models to advanced semantic inference or NL generation tasks (Wang et al. 2018). Recently, pre-trained LMs have been augmented with **external knowledge** from commonsense knowledge bases such as ConceptNet, which provides more explicit knowledge grounding and improves their performance on downstream tasks that require reasoning abilities. Wang et al. (2020b), for example, retrieve multi-hop knowledge paths from ConceptNet for fine-tuning LMs for multiple choice question answering. Chang et al. (2020) and Bosselut et al. (2021) incorporate knowledge paths from ConceptNet into pre-trained LMs for solving the SocialIQA task (Sap et al., 2019). However, all these approaches evaluate the effectiveness of integrating commonsense knowledge *indirectly* on downstream tasks, and do not explicitly evaluate the impact and relevance of knowledge for a specific system prediction. We address this shortcoming by generating and carefully evaluating statements that connect pairs of sentences as *explanations* of their underlying, *implicit knowledge* link. Closest to this aim is the task of **explanation generation**, which has received attention very recently. Wang et al. (2020a) propose the SemEval-2020 Task 4 (Subtask C), which is to generate an explanation for why a statement does not make sense, by way of a natural language statement. A comparison of the participating systems (cf. Perumal et al./Jon et al. 2020) shows that **pre-trained LMs** play a central role in the success of the top-performing systems, demonstrating that they contain commonsense information to a good extent. The success of models enriched with **knowledge from external sources** such as ConceptNet furthermore shows that additional knowledge supports the generation of commonsense explanations. However, there is still a large gap between systems and human performance.

Pre-trained LMs enhanced with commonsense knowledge have also been the models of choice for other **text generation** tasks, e.g. dialogue generation (Zhou et al., 2018), story ending generation (Guan et al., 2020), or abductive NLI (Ji et al., 2020b). While these models aim at generating explanations for a *single* statement, or *completing* a given sequence of sentences, we investigate how to make use of LMs to generate a sentence that fills in implicit knowledge *between* two sentences.

Constraining LMs. Recent work addresses how to control content in LM text generation, while maintaining fluency, coherence and plausibility of the generated text. Lin et al. (2020) explore how to generate a coherent and plausible situation description given an unordered **set of concepts** as input, and find that even pre-trained LMs (BART, T5) fine-tuned to this task cannot solve it: the generated sentences are grammatical, but highly implausible, lacking commonsense. This suggests that either the underlying LMs, or input constraints for generation need to incorporate commonsense knowledge. Orbach and Goldberg (2020) attempt to control the content when generating longer stories by **specifying facts** the story needs to include. They propose a plan-and-cloze model that first creates a cloze template, placing input facts at fixed positions in the output. In the cloze step, the system expands the fact tokens into complex sentences that complete the story. While uni-directional LMs such as GPT-2 or BART generate fluent text but do not well adhere to the desired content, the fine-tuned multi-directional XLNet outputs coherent text and adheres to the facts.

While none of the above works incorporate external knowledge to guide generation, Ji et al. (2020a) perform explanation generation for *single statements*, using ConceptNet background knowledge. The model selects concepts from the statement, retrieves connecting paths from ConceptNet, and selects **bridge concepts** from a subgraph. A pre-

[1] The code for our proposed approach can be found here: https://github.com/Heidelberg-NLP/LMs4Implicit-Knowledge-Generation.

trained decoder generates the explanation, using as input the statement and top-ranked concepts from the subgraph. In our work we also select concepts from texts, but *dynamically* generate commonsense knowledge paths as constraints. Importantly, we aim to generate coherent explanations *in-between* sentences – a challenge for uni-directional LMs.

3 Knowledge-constrained text generation

3.1 Task Definition and Approach

The task we tackle in this work is: given two contiguous sentences (*source sentences* S_1, S_2), generate an explanatory sentence (*target sentence* T) that explains the underlying, implicit information that connects them. We explore **different types of LMs** and their aptness for solving this task. We **fine-tune** them on existing or adapted datasets to inject relevant knowledge, and add *key concepts* or *connecting knowledge-paths* as **constraints** to achieve coherent and informative explanations.

3.2 Types of Language Models

We compare three types of LMs: **GPT-2** (Radford et al., 2019), an autoregressive model which generates the output sequence from left to right; **XLNet** (Yang et al., 2019b), a bidirectional generalized autoregressive LM; and **BART** (Lewis et al., 2019), a seq2seq model with a bidirectional masked encoder and a left-to-right decoder. While GPT-2 and BART generate the next tokens seeing only the left (previous) context, XLNet predicts the next tokens based on the left *and* right context, in a random order. GPT-2 is pre-trained on web pages from CommonCrawl, XLNet on CommonCrawl+ClueWeb (Callan et al., 2009), and BART on the CNN/DM summarization dataset (Hermann et al., 2015).

3.3 Fine-tuning LMs

Task-adapted Datasets for LM Fine-tuning. All chosen LMs are pre-trained on information that is *explicit* in text. To condition them to generate *implicit* information that connects sentences, we fine-tune them on datasets that include knowledge statements connecting contiguous sentence pairs. We create two such corpora, one based on Generics-KB (Bhakthavatsalam et al., 2020), which offers statements expressing generic knowledge; the other on e-SNLI (Camburu et al., 2018), which comprises explanations of inferential commonsense knowledge. Each data instance contains two source sentences S_1, S_2, a target sentence T, and two key concepts c_1, c_2 which we extract from the original data as described below. For examples see Table 1.

Generics-KB contains naturally occurring generic sentences crawled from the web using linguistic rules and BERT-based scoring. It is rich in high-quality statements that express generic knowledge. Each generic sentence occurs in its surrounding context (1-5 sents before/after), hence each instance forms a triple consisting of the context before (C_b), the generic sentence (GS) and the context after (C_a). We collect all instances where a phrase p_1 (NP, VP, ADJP or ADVP) from GS also occurs in C_b, and another phrase p_2 from GS occurs in C_a. For each instance we extract the sentence containing p_1 and the one containing p_2 as our source sentences S_1, S_2; GS as our target sentence T; and p_1 and p_2 as key concepts c_1, c_2.

e-SNLI is an extension of the SNLI dataset (Bowman et al., 2015), additionally annotated with explanations: Given a premise-hypothesis pair and the relation between them (entailment, contradiction, or neutral), annotators added natural language sentences that explain why the pair is in the relation. Annotators had to mark essential key phrases for the relation in premise and hypothesis, and had to formulate explanations that employ these key phrases. For fine-tuning and testing our models, we consider all instances labelled with entailment and contradiction relations (but do not include the labels in fine-tuning). We interpret premise and hypothesis as our source sentences S_1 and S_2, the explanation as our target sentence T, and the marked key phrases as our key concepts c_1 and c_2.

In- and Out-Of-Domain Test Sets. We test the resulting models *in-domain* – on testsets from our fine-tuning corpora; and *out-of-domain* – on the **IKAT** dataset (Becker et al., 2020), which is based on the argumentative Microtexts Corpus (Peldszus and Stede, 2015). For all sentence pairs S_1 and S_2 that are adjacent or argumentatively related, annotators added the implicit knowledge that connects them, using simple sentences, which we use as targets T. They also marked two key phrases in each implicit knowledge sentence, where in most cases one key phrase appears in the first source sentence, and the other in the second – which we interpret as key concepts c_1 and c_2 in our approach.

3.4 Constraining Explanation Generation

Our hypothesis is that *unconditioned* generation may not be sufficient to produce statements carry-

Dataset	Source Sentence 1	Source Sentence 2	Target Sentence	Key Concepts	Paths
G-KB	The patient is assessed for pain every 4 hours.	Combination of treatments are needed for effective pain management.	Pain management is provided for patients in pain.	pain management, patients	pain management USEDFOR patients
eSNLI	The city has a lot of people in it.	It is busy city that looks like New York City	A city that is busy has a lot of people in it.	a lot of people, busy city	busy city HASPROPERTY a lot of people
IKAT	Education and training are fundamental rights which the state must provide.	If a university lacks the fundings, sponsors must be found.	Education and training must be funded.	education and training, fundings	education and training HAS PREREQUISITE fundings

Table 1: **Source sentence pairs** and **target sentences** (reference) from our three datasets, with marked **key concepts** and automatically predicted knowledge **paths** between them.

Gen-KB	BL	*Patients often report back to the clinic with a worsening pain condition within one to two hours of first assessment.*
	+c	*Patients often have few if any symptoms at first, but pain becomes less intense and less frequent in coming hours.*
	+p	*Patients are admitted to the hospital with moderate to high intensity pain.*
e-SNLI	BL	*A busy city that looks like new york city has a lot of people in it, so the city has to have a lot to people in the city.*
	+c	*The city has a lot of people in it because it is a busy city.*
	+p	*A busy city implies that there are a lot of people in the city.*
IKAT	BL	*The state and society must be found if a university lacks the funds to provide education and training.*
	+c	*The state and the society must pay for education and training if the university lacks the funds.*
	+p	*If a university lacks the funds, it can not be providing education and training to its students.*

Table 2: **Example generations** for pairs from Tab. 1, from **BART**: w/o constraints or **constrained** w/ concepts (c) or paths (p).

ing relevant knowledge which explains the connection between two sentences. Hence we experiment with direct injection of constraints or triggers to guide the generation to emit meaningful and coherent implicit knowledge statements: We include (i) **key concepts** as offered by each dataset, since we expect them to direct the model towards concepts that are relevant for explaining how the two sentences are related. We also include (ii) relational knowledge between the key concepts as constraints, by establishing **multi-hop knowledge paths** between them. To this end we combine relation classification and target prediction models specifically adapted to ConceptNet. The two respective models are based on LMs fine-tuned on ConceptNet (Speer et al., 2017), a large network that represents commonsense facts.[2] We generate single- and multihop paths between key concepts from a sentence pair, and use these paths as constraints when generating target sentences. We expect the generated paths to provide useful relational information for the model. Example paths appear in Table 1.

4 Data and Experimental Setup

Datasets. We use the data from GenericsKB and e-SLNI for fine-tuning and testing models (in-domain), and IKAT for testing out-of-domain.[3] For statistics see Table 3. All instances contain two source sentences $S_{1,2}$, a target sentence T, and two key concepts $c_{1,2}$, where $c_1 \in S_1$, $c_2 \in S_2$, and $c_{1,2} \in T$. We experiment with $c_{1,2}$, and with paths p generated between c_1 and c_2 as constraints, which we establish as explained above.

Input Sequences. We build the input sequences by concatenating the source sentences S_1 and S_2, separated by a SEP token. When including key concepts $c_{1,2}$ or knowledge paths p as constraints, we append them to the input sequence right after S_1 and S_2, separated by a SEP token. Thus, the concepts and paths we use as constraints are encoded by the tokenizer of each language model together with the rest of the input sequence. Accordingly, our input sequences are structured as follows: S_1 <SEP> S_2 <SEP> $(c_1, c_2|p)$ <EOT> T.

Fine-tuning LMs. For LM fine-tuning, we append the target sentence to the input sequence, separated from the rest of the input by an EOT tag. GPT-2 and XLNet are trained to reconstruct the target sentence T. During inference, the models only see the source sentences, and constraints if

[2] Details about the models appear in the Appendix.

[3] In preliminary experiments we also tried to fine-tune our LMs on GenericsKB *and* e-SNLI together, which did not improve results compared to when using these datasets separately for fine-tuning – most likely because the datasets are very different from each other in terms of linguistic characteristics (e.g. sentence lengths and structure) and the covered topics.

	train	dev	test	eval-1	eval-2
G-KB	21,644	6,184	3,091	10	30
e-SNLI	18,160	2,028	1,002	10	30
IKAT	-	-	719	10	40

Table 3: Datasets: Nb. of **source sentence pairs** with associated implicit knowledge sentences, used for fine-tuning and testing; and subsets from test used in evaluations.

given, and they complete the input sequence by generating T. In contrast, BART encodes S_1 and S_2, and its decoder is trained to predict T based on the encoded source sentences.

We use the pre-trained models from HuggingFace Transformers (Wolf et al., 2019) and adapt them for fine-tuning on our customized training data. In order to generate compact sentences capturing the relevant implicit knowledge (instead of long explanations), we set a length limitation of 20 tokens for each generation. More details about our models are listed in the Appendix.

5 Evaluation and Results

This section presents an **in-depth evaluation** of the quality of generations from different model variants, and their ability of expressing implicitly conveyed knowledge. We design a **manual** evaluation setup covering various dimensions, and compare the results to several **automatic** evaluation metrics. We conduct evaluation *in-domain* on our customized test data; and *out-of-domain* on IKAT.

5.1 Manual Evaluation

Questions to Annotators.[4] To filter out source sentence pairs between which no implicit information is missing, we first ask the annotators for each source sentence pair if they are **implicitly connected** by some (unexpressed) piece of knowledge (*yes/no*). The annotators are then guided through follow-up questions covering four dimensions:
(1) **Grammaticality** – we ask if the generated sentence is grammatically correct, given the choices *correct, almost correct* (minor grammatical errors), and *incorrect* (major grammatical errors);
(2) **Coherence** – we ask if the generated sentence is logically and semantically consistent with respect to the two source sentences, given the choices *fully coherent, partly coherent*, or *incoherent*;
(3) **Content** – we ask if the generated sentence gives an **explanation** of the connection between the two source sentences, given the choices *yes*, *neutral* (if the generated sentence is related to the source sentences, but not in a clear logical relation), and *no* (if the sentence is misleading or contradictory in the context of the source sentences);[5] (4) **Comparison** to the annotated reference sentence [6] – we ask if the generated sentence is similar in meaning to the reference, given the choices *similar, partly similar*, or *not similar*. In addition, we ask if the reference sentence or the generated sentence is a more meaningful explanation of the implicit knowledge that connects the source sentences, or if both are equally meaningful explanations.

Annotation Setup. Our goal is to investigate which model variant is best suited for generating grammatically sound, coherent and meaningful explanations. We approach this question with two annotation rounds: In a first round we aim to determine which **model** is best suited for generating implicitly conveyed knowledge, and which **dataset** is best suited for fine-tuning the model for generating statements on *out-of-domain* test sets. In a second annotation round we aim to determine which types of **constraints** yield best results, now restricted to the best performing model and training setup, as determined in round one.

Annotator Agreement. Annotation was performed by two annotators with a background in computational linguistics. We measure IAA using Cohen's Kappa, combined over round one and two, and achieve an agreement of 95% on dimension 1, 80% on 2, 77% on 3, and on dimension 4 82% for the first and 78% for the second question. Remaining conflicts were resolved by an expert annotator.

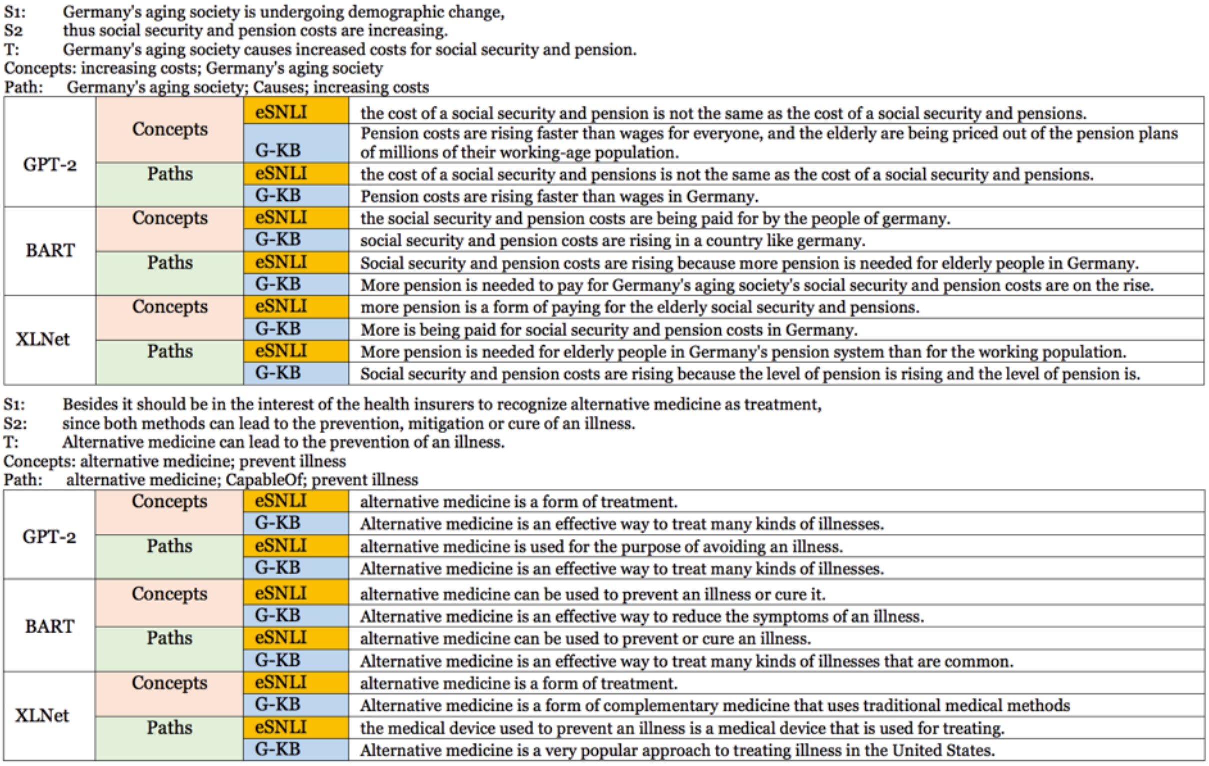
5.1.1 Best Model Type and Fine-Tuning Data

For the first annotation round we sample 10 source sentence pairs from each testset, hence 30 pairs overall, and the sentences generated by GPT-2, XLNet and BART for each instance, using concepts as

[4] The annotation manual together with example annotations can be found here: `https://github.com/Heidelberg-NLP/LMs4Implicit-Knowledge-Generation/blob/main/manual.pdf`

[5] The difference between dimension 2 and 3 is that with dimension 2 (*coherence*), we want to explore if the generated sentence semantically fits to the two given source sentences. We understand coherence together with Hobbs (1979) as the existence of specific knowledge relations that hold between concepts in a text (or discourse), such as *Cause-Effect, Condition*, or *Temporal Sequence*, cf. Wolf and Gibson (2004). These relations make the texts interpretable and informative and are motivated ultimately by the speaker's or writer's need to be understood (Hobbs, 1979). In contrast, when evaluating the *content* of the generated sentence in dimension 3, we want to discover if the sentence really *explains* the connection between the two source sentences.

[6] The reference sentence is *only* provided for Question 4.



S1: Germany's aging society is undergoing demographic change,
S2 thus social security and pension costs are increasing.
T: Germany's aging society causes increased costs for social security and pension.
Concepts: increasing costs; Germany's aging society
Path: Germany's aging society; Causes; increasing costs

GPT-2	Concepts	eSNLI	the cost of a social security and pension is not the same as the cost of a social security and pensions.
		G-KB	Pension costs are rising faster than wages for everyone, and the elderly are being priced out of the pension plans of millions of their working-age population.
	Paths	eSNLI	the cost of a social security and pensions is not the same as the cost of a social security and pensions.
		G-KB	Pension costs are rising faster than wages in Germany.
BART	Concepts	eSNLI	the social security and pension costs are being paid for by the people of germany.
		G-KB	social security and pension costs are rising in a country like germany.
	Paths	eSNLI	Social security and pension costs are rising because more pension is needed for elderly people in Germany.
		G-KB	More pension is needed to pay for Germany's aging society's social security and pension costs are on the rise.
XLNet	Concepts	eSNLI	more pension is a form of paying for the elderly social security and pensions.
		G-KB	More is being paid for social security and pension costs in Germany.
	Paths	eSNLI	More pension is needed for elderly people in Germany's pension system than for the working population.
		G-KB	Social security and pension costs are rising because the level of pension is rising and the level of pension is.

S1: Besides it should be in the interest of the health insurers to recognize alternative medicine as treatment,
S2: since both methods can lead to the prevention, mitigation or cure of an illness.
T: Alternative medicine can lead to the prevention of an illness.
Concepts: alternative medicine; prevent illness
Path: alternative medicine; CapableOf; prevent illness

GPT-2	Concepts	eSNLI	alternative medicine is a form of treatment.
		G-KB	Alternative medicine is an effective way to treat many kinds of illnesses.
	Paths	eSNLI	alternative medicine is used for the purpose of avoiding an illness.
		G-KB	Alternative medicine is an effective way to treat many kinds of illnesses.
BART	Concepts	eSNLI	alternative medicine can be used to prevent an illness or cure it.
		G-KB	Alternative medicine is an effective way to reduce the symptoms of an illness.
	Paths	eSNLI	alternative medicine can be used to prevent or cure an illness.
		G-KB	Alternative medicine is an effective way to treat many kinds of illnesses that are common.
XLNet	Concepts	eSNLI	alternative medicine is a form of treatment.
		G-KB	Alternative medicine is a form of complementary medicine that uses traditional medical methods
	Paths	eSNLI	the medical device used to prevent an illness is a medical device that is used for treating.
		G-KB	Alternative medicine is a very popular approach to treating illness in the United States.

Figure 1: Example generations for the IKAT test set, for all three models, fine-tuned on e-SNLI vs. GenericsKB, with concepts vs. paths as constraints.

constraints. For IKAT, we consider the sentences generated by each model fine-tuned on e-SNLI vs. GenericsKB. This sums up to 120 annotation samples (generated sentences).[7] In Fig. 1 we give example generations for IKAT, for all three model types, comparing fine-tuning on e-SNLI vs. GenericsKB; and constraining with concepts vs. with paths. More examples appear in the Appendix.

Results. For all 30 sentence pairs the annotators agreed that there is some implicit information connecting them. Table 4 displays the results of the first annotation round for the four dimensions described above. All three models are able to generate **grammatically correct** sentences (col. 1), with BART's generations scored as correct most often. BART also generates the most **coherent** sentences (col. 2), in-domain (e-SNLI and GenericsKB) and out-of-domain (IKAT), followed by XLNet. For dimension 3, which evaluates whether the generations are **meaningful explanations** of implicit knowledge connecting the source sentences (col. 3), only BART fine-tuned on e-SNLI gives satisfactory results (in-domain, when fine-tuned and tested on e-SNLI; *and* out-of domain, when fine-tuned on e-SNLI and tested on IKAT). Many of the generations from GPT-2 are judged as neutral (orange in Table 4) or misleading (red). The last two columns reflect the **comparison** of the generated vs. annotated **reference sentence** (dimension 4). BART's generations are overall rated as most similar to the reference sentence, especially when fine-tuned on e-SNLI (in- and out-of-domain), and are judged as better or equally good explanations compared to the reference sentences in 70% (e-SNLI, in-domain) and 50% (IKAT–e-SNLI, out-of-domain).

To summarize, according to our first round of evaluation, the BART model generates the most grammatical and coherent statements that are found to explain the connection between the source sentences best. They are also judged to be most similar to the reference sentence. When applied on out-of-domain testsets, BART performs best when fine-tuned on e-SNLI.

5.1.2 Best Constraints

While the first round of annotations used a relatively small set of 120 generated target sentences that helped us to determine BART as the best-suited model type, we now aim to deeper investigate the generations of BART to study the **effect of different types of constraints** on the quality of expla-

[7] 30 generated sents for e-SNLI and GenericsKB, resp. (10 source sents x 3 models), and 60 generated sents for IKAT (10 source sents x 3 models x 2 different fine-tuning datasets).

	DIMENSION	Grammaticality	Coherence	Explanation	Sim. to Reference	Gen. vs. Ref.
	CHOICES	Yes/Almost/No	Yes/Partly/No	Yes/Neutral/No	Yes/Partly/No	GS/Both/RS
GPT-2	e-SNLI	60/30/10	30/20/**50**	60/20/20	40/20/40	20/20/60
	G-KB	**100**/0/0	40/50/10	30/**70**/0	20/40/40	0/20/**80**
	IKAT - e-SNLI	70/10/20	20/30/**50**	20/**80**/0	20/60/20	0/40/60
	IKAT - G-KB	**100**/0/0	40/50/10	20/60/20	20/20/**60**	10/10/**80**
XLNet	e-SNLI	90/10/0	60/20/20	60/20/20	60/20/20	30/30/40
	G-KB	90/10/0	40/50/10	50/50/0	0/60/40	20/10/70
	IKAT - e-SNLI	80/20/0	**60**/20/20	50/40/10	30/60/10	0/40/60
	IKAT - G-KB	90/0/10	20/80/0	50/30/20	10/20/**70**	0/10/**90**
BART	e-SNLI	**100**/0/0	**100**/0/0	**100**/0/0	**80**/20/0	**40/30**/30
	G-KB	**100**/0/0	40/60/0	40/60/0	20/80/0	20/10/70
	IKAT - e-SNLI	90/0/0	**60**/40/0	**70**/20/10	**60**/30/10	**20/30**/50
	IKAT - G-KB	**100**/0/0	50/50/0	50/40/10	50/40/10	40/0/60

Table 4: Results of the 1st manual evaluation (in %). For all 10 source sentence pairs, each model generates a target sentence when fine-tuned and tested *in-domain* on (i) e-SNLI and (ii) GenericsKB; or *out-of-domain* testing on IKAT, when fine-tuned on (iii) e-SNLI or (iv) GenericsKB; with marked best/worst scores for in- and out-of domain testing.

nations. We provide our annotators with 70 new source sentence pairs (20 from e-SNLI, 20 from GenericsKB, 30 from IKAT), and three different targets per pair, generated by three model variants of BART: (i) a baseline fine-tuned *without* any knowledge constraints; (ii) BART fine-tuned using the *key concepts* as constraints; and (iii) BART fine-tuned using an automatically generated *commonsense knowledge path* between the key concepts as constraint. Since fine-tuning on e-SNLI has been determined as best suited for out-of-domain testing, we consider only generations from BART fine-tuned on e-SNLI for testing on IKAT. In our evaluation we consider the 70 sentence pairs and the respective sentence generations from Round 2, and the generations for the 30 source sentence pairs from the best performing model BART from Round 1, resulting in 100 sentence pairs, with three generations per pair.

Results. Similar to Round 1, for 98% of the source sentence pairs the annotators agreed that there is some implicit information connecting them.

Fig. 2 shows the results of the second round of evaluations, example generations appear in Table 2. We find that using **knowledge constraints improves the quality** of generations compared to the baseline without constraints, on all four dimensions: on each of our three test sets, generations are rated as more grammatical when constrained with concepts and paths (with GenericsKB as only exception); they are annotated as more coherent, and rated as better explanations of implicit knowledge. Knowledge constraints also lead to a higher similarity to the reference sentence on all three datasets, and sentences generated with knowledge constraints are more often rated as better explanations than the reference sentences. Overall we find that *knowledge paths* improve scores over the baseline *more than concepts* (a plus of 2–15 pp). The improvements are most significant for IKAT, where adding concepts boosts evaluation scores between 18 (Grammaticality) and 53 pp (Coherence), and adding paths by 20 (Grammaticality) and 55 pp (Coherence). The generations of BART, fine-tuned on e-SNLI, as shown in the first example in Fig. 1, demonstrate how the integration of paths as constraints can improve text generation even more than when only injecting key concepts. The path used as constraint is *Germany's aging society* CAUSES *increasing costs*. When constraining BART with key concepts, it generates *The social security and pension costs are being paid for by the people of Germany*, while the generation with the knowledge path as constraint is *Social security and pension costs are rising because more pension is needed for elderly people in Germany*). This shows that the relation CAUSES gives our model an important hint about the causal relation that is needed to explain the connection between the two given sentences.

To summarize, the results from our second evaluation round clearly show that constraints in form of relevant concepts and knowledge paths can help LMs for generating grammatically sound, coherent and meaningful explanations of the missing knowledge between sentences, especially when applied on out-of-domain test sets.

5.2 Automatic Evaluation

In our automatic evaluation setup, we apply a range of different evaluation metrics commonly applied in text generation tasks, which either measure the *similarity* to a reference sentence (in our case, the

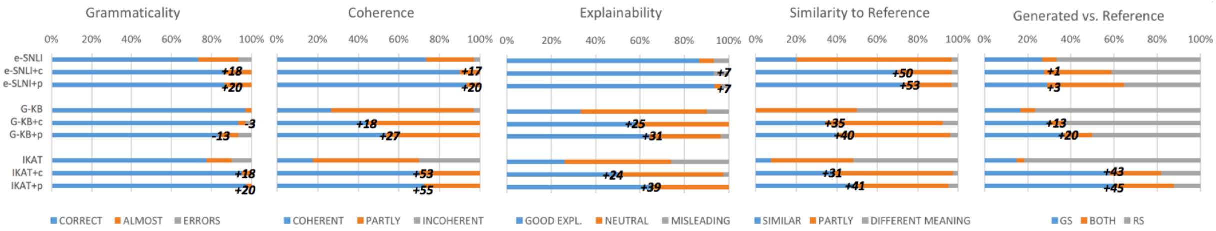


Figure 2: Results of 2^{nd} manual evaluation: comparing models constrained with concepts (+c) or paths (+p) against a baseline without constraints. We display improvements in percentage points (pp) for the best option (blue bar) per dimension.

generic sentences in GenericsKB, inference explanations in e-SNLI, or implicit knowledge statements in IKAT); or the *linguistic quality and diversity* of the generated sentence.

(i) **BLEU** (Papineni et al., 2002) and **ROUGE** (Lin, 2004) measure token overlap using ngrams. We apply BLEU-1 to measure precision and ROUGE-1 to measure recall based on unigrams;

(ii) **BERT-Score** (Zhang* et al., 2020) and **Sentence-BERT** (Reimers and Gurevych, 2019) compute semantic similarity scores for text sequences based on word or sentence representations. BERT-Score uses BERT's contextualized word embeddings to calculate a cross similarity score for each token in the generation with each token in the reference, while Sentence-BERT is fine-tuned on NLI and STS to predict the similarity of two sequences. For BERT-Score we report F1 scores; for Sentence-BERT we average the similarity scores obtained for the generated vs. reference sentences.

(iii) **S2Match** (Opitz et al., 2020) is an AMR graph matching metric, which measures the overlap of the AMR semantic graphs that we construct from the reference and generated sentence using Cai and Lam (2020)'s parser, and reports accuracy;

(iv) **Distinct-N** (Li et al., 2015) and **GRUEN** (Zhu and Bhat, 2020) are *reference-free* metrics that only consider properties of the generated sentence. Distinct-N measures the diversity of a sentence by focusing on the number of distinct unigrams (Distinct-1) and bigrams (Distinct-2); GRUEN evaluates the linguistic quality of a sentence in terms of grammaticality, non-redundancy, and structure.

In a **preliminary experiment** based on the *complete test sets* of Generics-KB, e-SNLI and IKAT (cf. Table 3) we first investigate which **model** generates sentences that are most similar to the reference sentence (using reference-based metrics), or which show highest linguistic quality and diversity (using reference-free metrics); and which **dataset** is best suited for fine-tuning the models for generating statements on *out-of-domain* test sets (here, IKAT). Results and detailed analysis of this experiment appear in our Appendix. We find that deciding which **model** performs best depends a lot on the chosen similarity metric, but overall we don't see the clear superiority of the BART model (nor the inferiority of GPT-2) that we determined through manual evaluation. While in Dimension 4 of the manual evaluation setup (where annotators judged whether generated and reference sentence express the same or similar meaning), BART was clearly rated as the best performing model, this is not reflected in the automatic evaluation scores. Among all metrics only **SentenceBERT**, giving highest scores to BART, followed by XLNet, aligns with our observations from manual evaluation. However, our other observation from manual evaluation – that **e-SNLI** is the most appropriate dataset for fine-tuing LMs for out-of-domain testing — aligns with the scores obtained by automatic evaluation metrics (for details, cf. Appendix).

	BLEU-1	ROU-1	S2M	BERT	S-BERT	dist1	dist2	GRUEN
e-SNLI	7.27	0.4	0.34	0.89	0.56	0.72	0.58	0.63
e-SNLI+c	12.71	0.47	0.38	0.90	0.63	0.75	0.66	0.63
e-SNLI+p	9.51	0.48	0.39	0.89	0.65	0.76	0.67	0.66
G-KB	1.22	0.15	0.31	0.88	0.53	0.71	0.62	0.82
G-KB+c	1.58	0.18	0.32	0.88	0.54	0.72	0.66	0.83
G-KB+p	1.14	0.17	0.31	0.89	0.56	0.73	0.67	0.80
IKAT	4.6	0.22	0.33	0.88	0.49	0.70	0.64	0.66
IKAT+c	6.06	0.31	0.42	0.90	0.63	0.72	0.67	0.71
IKAT+p	7.23	0.33	0.46	0.91	0.64	0.74	0.70	0.76

Table 5: Automatic similarity scores for generations of best performing model BART, w/o constraints or with concepts/paths as constraints. Adding concepts and paths improves scores *in-domain* (e-SNLI and Generics-KB), and *out-of-domain* (IKAT finetuned on e-SLNI).

We next analyse which types of **constraints** improve generation, focusing on the *BART* model, which has shown to be best for generating implicit knowledge statements in our manual evaluation setup. Our automatic evaluation is based on the same *subset* of source sentence pairs used for the second round of manual annotations (cf. Table 3), and we again compare generations without constraints to conditioning on key concepts or knowledge paths.[8] Results are displayed in Table 5. We observe that for all metrics, scores increase when constraining LMs with concepts or knowledge paths, with BLEU and S2Match scores for GenericsKB as only exceptions. As in manual evaluation (Fig. 1), we find that improvements are most significant for IKAT. The observed improvements may in part be traced back to increased word overlap due to key concepts being used as constraints. Yet we also observe that automatically generated knowledge paths between these concepts improve scores additionally – according to reference-based metrics (showing that generations become more similar to references), and reference-free metrics (showing improvement of the linguistic quality and diversity of generations). This points to the fact that constraining LMs with automatically generated relational knowledge is a promising step towards generating grammatically correct and meaningful implicit knowledge statements.

6 Discussion

Limitations of Automatic Evaluation Metrics for Text Generations. Concluding, we pinpoint two important limitations of automatic text generations metrics – especially reference-based ones: Besides well-known issues regarding the reliability, interpretability and biases of such metrics (Callison-Burch et al., 2006), scores are mostly obtained by comparing generations against a single reference, which is – here, as in other generation tasks – often only *one* among *several* valid options. For the task of reconstructing implicit information, Becker et al. (2017) show that annotators often propose different valid sentences for filling knowledge gaps in argumentative texts. For our setting this means that a generated sentence may be a relevant explicitation of implicit information, even if *not* similar to the reference. Such cases are poorly or not at all captured by automatic similarity metrics. An exception we found is SentenceBERT, which is based on sentence representations, and which aligned reasonably well with insights from our manual evaluation. Still, automatic evaluation of text generations needs to be considered with caution, and should always be accompanied by manual evaluation.

Our Implicitness Assumption. Our experiments are based on the underlying assumption that usually some information between pairs of sentences stays implicit, which has been confirmed empirically for our datasets: Our annotators stated for 100% (first round) and 98% (second round) of all sentence pairs that they are implicitly connected by some unexpressed piece of knowledge. However, we did not specifically address the cases of sentence pairs between which no implicit information is missing (even though these cases are rare), nor did we investigate how our models would perform when provided with sentence pairs that are *not* related (arbitrary pairs). For a real-world application, both aspects would be considerable.

7 Conclusion

In this work we propose an approach for generating statements that explicate implicit knowledge connecting sentences in text, using pre-trained LMs. We show that despite their great success in many NLP downstream tasks, LMs need to be well equipped and carefully guided for the challenging task of reconstructing implicit knowledge, to ensure that they convey the missing, implicit information that connects sentences in text. We refine different pre-trained LMs by fine-tuning on specifically prepared corpora that we enrich with implicit information, filled in between sentences, and explore methods of constrained language generation, guiding the models by way of relevant concepts and connecting commonsense knowledge paths.

While most current automatic NLG metrics are not sufficient to evaluate this challenging task, our in-depth evaluation of the quality of generations from different model variants shows that the BART model, which attends over its full input when generating text, yields most informative and relevant explanations. We also establish that e-SNLI, being focused on the NLI task, is best suited for conditioning LMs for our task, especially for out-of domain settings. Finally, by providing the LMs with relevant connecting key concepts as constraints, and further by connecting commonsense knowledge paths, we achieve generation of coherent and grammatically sound sentences that – according to manual evaluation – can explicate the implicit knowledge that connects sentence pairs in texts – for in-domain and out-of-domain test data.

[8] The automatic evaluation scores for the complete test sets, which confirm our findings from the subset of the second annotation round, appear in the Appendix.

References

Maria Becker, Katharina Korfhage, and Anette Frank. 2020. Implicit Knowledge in Argumentative Texts: An Annotated Corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC)*, pages 2316–2324, Marseille, France.

Maria Becker, Michael Staniek, Vivi Nastase, and Anette Frank. 2017. Enriching Argumentative Texts with Implicit Knowledge. In *Applications of Natural Language to Data Bases (NLDB) - Natural Language Processing and Information Systems*, Lecture Notes in Computer Science. Springer.

Maria Becker, Michael Staniek, Vivi Nastase, and Anette Frank. 2019. Assessing the difficulty of classifying ConceptNet relations in a multi-label classification setting. In *RELATIONS - Workshop on meaning relations between phrases and sentences*, Gothenburg, Sweden. Association for Computational Linguistics.

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. GenericsKB: A Knowledge Base of Generic Statements. In *Arxiv Preprint*.

Antoine Bosselut, Ronan Le Bras, , and Yejin Choi. 2021. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Celikyilmaz Asli, and Choi Yejin. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *ACL*, pages 4762–4779.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Deng Cai and Wai Lam. 2020. Amr parsing via graph-sequence iterative inference.

Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. 2009. Clueweb09 data set.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31, pages 9539–9549. Curran Associates, Inc.

Ting-Yun Chang, Yang Liu, Karthik Gopalakrishnan, Behnam Hedayatnia, Pei Zhou, and Dilek Hakkani-Tur. 2020. Incorporating commonsense knowledge graph in pretrained models for social commonsense tasks. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 74–79, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*.

Jerry R. Hobbs. 1979. Coherence and coreference*. *Cognitive Science*, 3(1):67–90.

Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, and Minlie Huang. 2020a. Generating commonsense explanation by extracting bridge concepts from reasoning paths. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 248–257, Suzhou, China. Association for Computational Linguistics.

Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020b. Language generation with multi-hop reasoning on commonsense knowledge graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 725–736, Online. Association for Computational Linguistics.

Josef Jon, Martin Fajcik, Martin Docekal, and Pavel Smrz. 2020. BUT-FIT at SemEval-2020 task 4: Multilingual commonsense. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 374–390, Barcelona (online). International Committee for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *CoRR*, abs/1510.03055.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

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

Juri Opitz, Anette Frank, and Letitia Parcalabescu. 2020. Amr similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, 8(0):522–538.

Eyal Orbach and Yoav Goldberg. 2020. Facts2Story: Controlling text generation by key facts. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2329–2345, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

Andreas Peldszus and Manfred Stede. 2015. An annotated corpus of argumentative microtexts. In *Proceedings of the First European Conference on Argumentation*.

Anandh Perumal, Chenyang Huang, Amine Trabelsi, and Osmar Zaïane. 2020. Ana at semeval-2020 task 4: multi-task learning for commonsense reasoning (union).

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *ArXiv*, abs/1904.09728.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *AAAI*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020a. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 307–321, Barcelona (online). International Committee for Computational Linguistics.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020b. Connecting the dots: A knowledgeable path generator for commonsense question answering. *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4129–4140.

Florian Wolf and Edward Gibson. 2004. Representing discourse coherence: A corpus-based analysis. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 134–140, Geneva, Switzerland. COLING.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, J. Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*.

Wanzheng Zhu and Suma Bhat. 2020. GRUEN for evaluating linguistic quality of generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 94–108, Online. Association for Computational Linguistics.

A Training Details

Finetuning Language Models. Details about the models and fine-tuning procedure as well as the running time for one batch are listed in Table 6. We fine-tuned all models with 2 GPUs on 3 epochs. Our training batch size is 8 as suggested by the HuggingFace's Transformers framework (Wolf et al., 2019). GPT-2 is the lightest one of our three models and takes 4 hours for fine-tuning on our e-SNLI and

GenericsKB datasets, respectively, while BART requires 8 hours, and XLNet around 20 hours (due to its permutation procedure) for the same data.

Limiting Length of Generations. In order to generate compact sentences capturing the relevant implicit knowledge (instead of long explanations), we set a length limitation of 20 tokens for each generation. In the left-to-right decoding procedure of GPT-2 and BART, the generation can be stopped earlier than 20 tokens, when the model predicts an EOT token. Thus, both GPT-2 and BART models can predict complete sentences of up to 20 tokens due to the autoregressive decoder. In contrast, XLNet has a permutation language modeling mechanism and predicts the next tokens based on the previous and next tokens. Its generations usually don't contain a significant EOT token. predicted target sequence of tokens in a post-processing step by cutting it after a generated comma (,).

Maximum Sequence Lengths. Our customized train sets have different maximum sequence lengths: e-SNLI has a maximum sequence length of 80 tokens including the target sentence, while GenericsKB has up to 140 tokens per sequence.

B Establishing Knowledge Paths for Constraining Text Generation

For dynamically establishing connections between the key concepts from two source sentences, we combine two model types: COREC-LM (Becker et al., 2019), an open-world multi-label relation classifier enhanced with a pretrained language model, that predicts *relation types* between two given concepts – for establishing direct connections between concepts; and COMET (Bosselut et al., 2019), a pretrained transformer model that learns to generate *target concepts* given a source concept and a relation, for generating multihop paths. By combining the generations of these models, we generate single- and multihop paths between key concepts c_1, c_2 from a sentence pair, and use these paths as constraints when generating target sentences. We are able to retrieve paths for 86.2% of all key concept pairs from GenericsKB, respectively, for 30.2% from e-SNLI and for 44.2% from IKAT. The differences can be explained by the fact that while the key concepts in GenericsKB are extracted phrases (NPs, VPs, ADJPs and ADVPs), the key concepts in e-SNLI and IKAT are manually labelled, and thus are often very specific and contain nested phrases (e.g. *leans over a pickup truck* (e-SNLI)). Therefore, it is more difficult to predict a relation or path between them. When we experiment with paths as constraints; for all instances where no path could be established between the key concepts, we only use the key concepts as constraints.

C Automatic Evaluation of the Complete Test Sets

As mentioned in Section 5.2 of our main paper, in a preliminary study based on the **complete test sets** of Generics-KB, e-SNLI and IKAT, we investigate which **model** generated sentences that are most similar to the reference sentence, or which show highest linguistic quality and diversity; and which **dataset** is best suited for finetuning the models for generating statements on *out-of-domain* test sets (here, IKAT). Results for this first analysis appear in Table 7. For metrics that measure token overlap (**BLEU** and **ROUGE**), highest scores are obtained when finetuning and testing on e-SNLI, which can be traced back to frequently used linguistic patterns (e.g., *x implies y*, or *x is the same as y*) that occur in train and test sets of e-SNLI. The reference-free metrics **Distinct** and **GRUEN** that measure diversity and non-redundancy, therefore yield higher scores when models are finetuned on the more diverse GenericsKB data, for both in- and out-of-domain testing. The AMR metric **S2Match** gives higher scores on e-SNLI than GenericsKB in in-domain testing, and finetuning on e-SNLI yields higher S2Match scores for out-of-domain testing on IKAT. This also aligns with the sentence representation based metric **SentenceBERT**. **BertScore**, finally, is not at all discriminative – it yields uniformly high scores for each model and configuration, ranging only between .88 and .9.

We also find that the scores differ considerably for **in-domain** vs. **out-of-domain** testing: results on IKAT are lower compared to testing on e-SNLI or GenericsKB according to all reference-based metrics, while we observe the opposite for the reference-free metrics.

We next analyse on the complete test set which types of **constraints** improve generation, focusing on the *BART* model, which has shown to be best for generating implicit knowledge statements in our manual evaluation setup. The automatic evaluation scores for the complete test sets are displayed in Table 8 and confirm our findings from the subset of the second annotation round, as presented in

Pretrained model ID	**Model details**	**Parameters**	**Time in s** (seq length = 80)	**Time in s** (seq length = 140)
gpt2	12-layer, 768-hidden, 12-heads	117M	0.039	0.056
xlnet-large-case	24-layer, 1024-hidden, 16-heads	340M	0.166	0.297
facebook/bart-large-cnn	24-layer, 1024-hidden, 16-heads	406M	0.075	0.116

Table 6: Benchmarks of the used pre-trained models.

Section 5.2 of our main paper.

D Example Generations

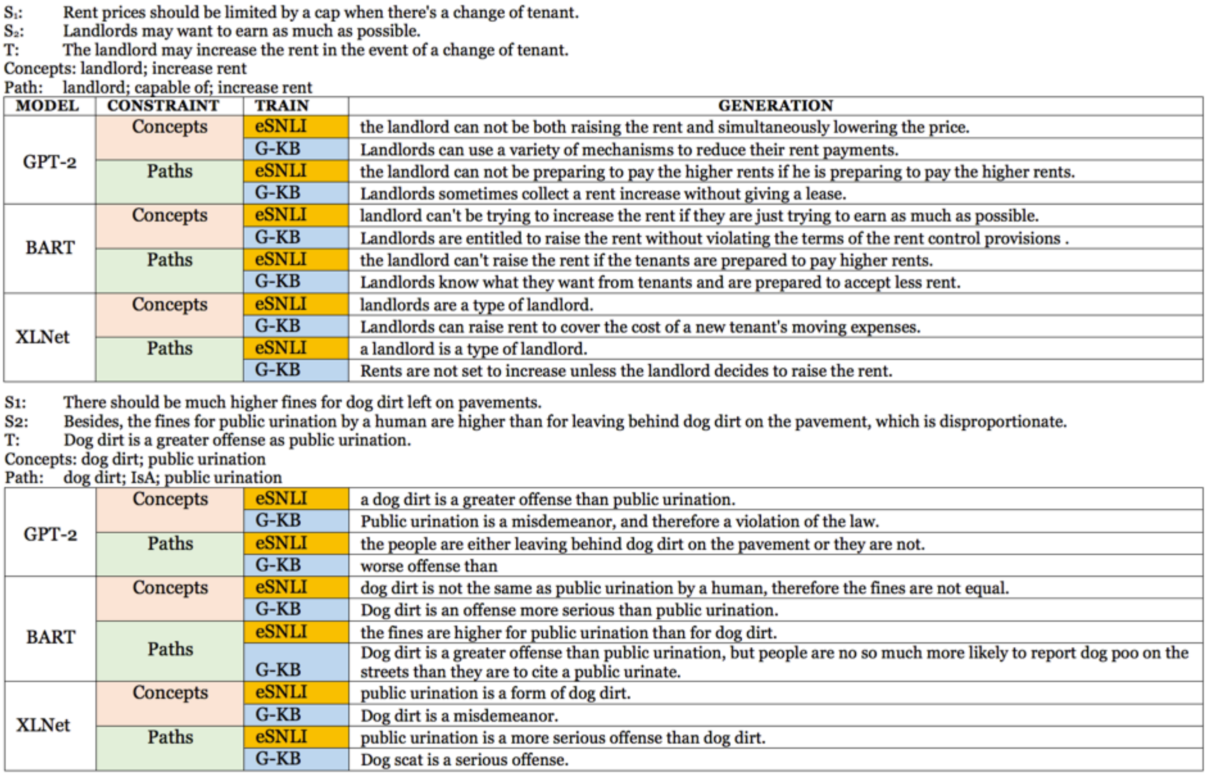
In addition to the examples shown in our main paper, in Fig. 1 we give some more example generations for the IKAT test set, for all three model types, comparing finetuning on e-SNLI vs. GenericsKB; and constraining with concepts vs. with paths.

TEST	TRAIN	BLEU-1	ROU-1	S2M	BERT	S-BERT	dist1	dist2	GRUEN
GPT-2									
G-KB	G-KB	5.3	.2	.33	.88	.5	.95	.89	.79
e-SNLI	e-SNLI	14.9	.46	.44	.89	.58	.91	.86	.52
IKAT	G-KB	2.9	.19	.3	.88	.45	.96	.85	.78
IKAT	e-SNLI	4.7	.26	.37	.89	.51	.88	.86	.64
XLNet									
G-KB	G-KB	6.6	.27	.36	.89	.53	.92	.87	.74
e-SNLI	e-SNLI	10.7	.43	.38	.89	.59	.88	.85	.58
IKAT	G-KB	4.2	.22	.34	.9	.48	.97	.88	.79
IKAT	e-SNLI	10.5	.33	.42	.9	.56	.9	.85	.69
BART									
G-KB	G-KB	5.2	.27	.35	.89	.57	.86	.93	.75
e-SNLI	e-SNLI	10.7	.44	.42	.89	.61	.81	.91	.59
IKAT	G-KB	2.37	.22	.3	.88	.53	.88	.93	.80
IKAT	e-SNLI	3.92	.29	.38	.9	.58	.87	.93	.71

Table 7: Automatic Similarity scores computed for the generations of all models, on the *complete test sets*. We compare the impact of (i) model types and (ii) data used for finetuning (train), in-domain (GenericsKB and e-SNLI) and out-of-domain (IKAT).

	BLEU-1	ROU-1	S2M	BERT	S-BERT	dist1	dist2	GRUEN
e-SNLI	7.36	0.37	0.36	0.88	0.54	0.77	0.89	0.59
e-SNLI+c	10.73	0.44	0.42	0.89	0.61	0.81	0.91	0.59
e-SNLI+p	11.71	0.44	0.43	0.89	0.62	0.84	0.92	0.59
G-KB	5.21	0.23	0.32	0.88	0.55	0.86	0.93	0.75
G-KB+c	5.2	0.27	0.35	0.89	0.57	0.86	0.93	0.75
G-KB+p	5.4	0.28	0.35	0.89	0.58	0.87	0.93	0.75
IKAT	2,74	0.19	0.29	0.87	0.43	0.86	0.92	0.67
IKAT+c	3.92	0.28	0.38	0.89	0.56	0.87	0.92	0.7
IKAT+p	4.84	0.3	0.4	0.9	0.57	0.9	0.93	0.72

Table 8: Automatic similarity scores for generations of best performing model BART on the *complete test sets*, w/o constraints or with concepts/paths as constraints. Adding concepts and paths improves scores *in-domain* (e-SNLI and GenericsKB), and *out-of-domain* (IKAT finetuned on e-SLNI).



S_1: Rent prices should be limited by a cap when there's a change of tenant.
S_2: Landlords may want to earn as much as possible.
T: The landlord may increase the rent in the event of a change of tenant.
Concepts: landlord; increase rent
Path: landlord; capable of; increase rent

MODEL	CONSTRAINT	TRAIN	GENERATION
GPT-2	Concepts	eSNLI	the landlord can not be both raising the rent and simultaneously lowering the price.
		G-KB	Landlords can use a variety of mechanisms to reduce their rent payments.
	Paths	eSNLI	the landlord can not be preparing to pay the higher rents if he is preparing to pay the higher rents.
		G-KB	Landlords sometimes collect a rent increase without giving a lease.
BART	Concepts	eSNLI	landlord can't be trying to increase the rent if they are just trying to earn as much as possible.
		G-KB	Landlords are entitled to raise the rent without violating the terms of the rent control provisions .
	Paths	eSNLI	the landlord can't raise the rent if the tenants are prepared to pay higher rents.
		G-KB	Landlords know what they want from tenants and are prepared to accept less rent.
XLNet	Concepts	eSNLI	landlords are a type of landlord.
		G-KB	Landlords can raise rent to cover the cost of a new tenant's moving expenses.
	Paths	eSNLI	a landlord is a type of landlord.
		G-KB	Rents are not set to increase unless the landlord decides to raise the rent.

S1: There should be much higher fines for dog dirt left on pavements.
S2: Besides, the fines for public urination by a human are higher than for leaving behind dog dirt on the pavement, which is disproportionate.
T: Dog dirt is a greater offense as public urination.
Concepts: dog dirt; public urination
Path: dog dirt; IsA; public urination

GPT-2	Concepts	eSNLI	a dog dirt is a greater offense than public urination.
		G-KB	Public urination is a misdemeanor, and therefore a violation of the law.
	Paths	eSNLI	the people are either leaving behind dog dirt on the pavement or they are not.
		G-KB	worse offense than
BART	Concepts	eSNLI	dog dirt is not the same as public urination by a human, therefore the fines are not equal.
		G-KB	Dog dirt is an offense more serious than public urination.
	Paths	eSNLI	the fines are higher for public urination than for dog dirt.
		G-KB	Dog dirt is a greater offense than public urination, but people are no so much more likely to report dog poo on the streets than they are to cite a public urinate.
XLNet	Concepts	eSNLI	public urination is a form of dog dirt.
		G-KB	Dog dirt is a misdemeanor.
	Paths	eSNLI	public urination is a more serious offense than dog dirt.
		G-KB	Dog scat is a serious offense.

Figure 3: Example generations for IKAT, for all three models, finetuned on e-SNLI vs. GenericsKB, with concepts vs. paths as constraints.