Language Models for Lexical Inference in Context

Martin Schmitt[1] and **Hinrich Schütze**[2]
Center for Information and Language Processing (CIS)
LMU Munich, Germany
[1]martin@cis.lmu.de [2]inquiries@cislmu.org

Abstract

Lexical inference in context (LIiC) is the task of recognizing textual entailment between two very similar sentences, i.e., sentences that only differ in one expression. It can therefore be seen as a variant of the natural language inference task that is focused on lexical semantics. We formulate and evaluate the first approaches based on pretrained language models (LMs) for this task: (i) a few-shot NLI classifier, (ii) a relation induction approach based on handcrafted patterns expressing the semantics of lexical inference, and (iii) a variant of (ii) with patterns that were automatically extracted from a corpus. All our approaches outperform the previous state of the art, showing the potential of pretrained LMs for LIiC. In an extensive analysis, we investigate factors of success and failure of our three approaches.[1]

1 Introduction

Lexical inference (LI) denotes the task of deciding whether or not an entailment relation holds between two lexical items. It is therefore related to the detection of other lexical relations like hyponymy between nouns (Hearst, 1992), e.g., *dog* ⇒ *animal*, or troponymy between verbs (Fellbaum and Miller, 1990), e.g., *to traipse* ⇒ *to walk*. Lexical inference in context (LIiC) adds the problem of disambiguating the pair of lexical items in a given context before reasoning about the inference question. This type of LI is particularly interesting for entailments between verbs and verbal expressions because their meaning – and therefore their implications – can drastically change with different arguments. Consider, e.g., *run* ⇒ *lead* in a PERSON / COMPANY context ("Bezos runs Amazon") vs. *run* ⇒ *execute* in a COMPUTER / SOFTWARE context ("My mac runs macOS"). LIiC is thus also closely related to the task of natural language inference (NLI) – also called recognizing textual entailment (Dagan et al., 2013) – and can be seen as a focused variant of it. Besides the important use case of evaluating NLI systems, this kind of predicate entailment has also been shown useful for question answering (Schoenmackers et al., 2010), event coreference (Shwartz et al., 2017; Meged et al., 2020), and link prediction in knowledge graphs (Hosseini et al., 2019).

Despite its NLI nature, previous systems for LIiC have primarily been models of lexical similarity (Levy and Dagan, 2016) or models based on verb argument inclusion (Hosseini et al., 2019). The reason is probably that supervised NLI models need large amounts of training data, which is unavailable for LIiC, and that systems trained on available large-scale NLI benchmarks (e.g., Williams et al., 2018) have been reported to insufficiently cover lexical phenomena (Glockner et al., 2018; Schmitt and Schütze, 2019).

Recently, transfer learning has become ubiquitous in NLP; Transformer (Vaswani et al., 2017) language models (LMs) pretrained on large amounts of textual data (Devlin et al., 2019a; Liu et al., 2019) form the basis of a lot of current state-of-the-art models. Besides zero- and few-shot capabilities (Radford et al., 2019; Brown et al., 2020), pretrained LMs have also been found to acquire factual and relational knowledge during pretraining (Petroni et al., 2019; Bouraoui et al., 2020). The entailment relation certainly stands out among previously explored semantic relations – such as the relation between a country and its capital – because it is very rarely stated explicitly and often involves reasoning about both the meaning of verbs and additional knowledge (Schmitt and Schütze, 2019). It is unclear whether implicit clues during pretraining are enough to learn about LIiC and what the best way is to harness any such implicit knowledge.

Regarding these questions, we make the follow-

[1] Our code is publicly available: https://github.com/mnschmit/lm-lexical-inference

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1267–1280
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

ing contributions: (1) This work is the first to explore the use of pretrained LMs for the LIiC task. (2) We formulate three approaches and evaluate them using the publicly available pretrained RoBERTa LM (Liu et al., 2019; Wolf et al., 2019): (i) a few-shot NLI classifier, (ii) a relation induction approach based on handcrafted patterns expressing the semantics of lexical inference, and (iii) a variant of (ii) with patterns that were automatically extracted from a corpus. (3) We introduce the concept of antipatterns, patterns that express non-entailment, and evaluate their usefulness for LIiC. (4) In our experiments on two established LIiC benchmarks, Levy/Holt's dataset (Levy and Dagan, 2016; Holt, 2018) and SherLIiC (Schmitt and Schütze, 2019), all our approaches consistently outperform previous work, thus setting a new state of the art for LIiC. (5) In contrast to previous work on relation induction (Bouraoui et al., 2020), automatically retrieved patterns do not outperform handcrafted ones for LIiC. A qualitative analysis of patterns and errors identifies possible reasons for this finding.

2 Related Work

Lexical inference. There has been a lot of work on lexical inference for nouns, notably hypernymy detection, resulting in a variety of benchmarks (Kotlerman et al., 2010; Kiela et al., 2015) and methods (Shwartz et al., 2015; Vulić and Mrkšić, 2018). Although there has been work on predicate entailment before (Lin and Pantel, 2001; Lewis and Steedman, 2013), Levy and Dagan (2016) were the first to create a general benchmark for evaluating entailment between verbs. In their evaluation, neither resource-based approaches (Pavlick et al., 2015; Berant et al., 2011) nor vector space models (Levy and Goldberg, 2014) achieved satisfying results. Holt (2018) later published a re-annotated version, which was readily adopted by later work. Hosseini et al. (2018) put global constraints on top of directed local similarity scores (Weeds and Weir, 2003; Lin, 1998; Szpektor and Dagan, 2008) based on distributional features of the predicates. Hosseini et al. (2019) replaced these scores by transition probabilities in a bipartite graph where edge weights are computed by a link prediction model.

When Schmitt and Schütze (2019) created the SherLIiC benchmark, they also mainly focused on resource- and vector-based models for evaluation. Their best model combines general-purpose word2vec representations (Mikolov et al., 2013) with a vector representation of the arguments that co-occur with a predicate.

All these works (i) base the probability of entailment validity on the similarity of the verbs and (ii) compute this similarity via (expected) co-occurrence of verbs and their arguments. Our work differs in that our models solely reason about the sentence surface in an end-to-end NLI task without access to previously observed argument pairs. This is possible because our models have learned about these surface forms during pretraining.

Patterns and entailment. Pattern-based approaches have long been known for hypernymy detection (Hearst, 1992). Recent work combined them with vector space models (Mirkin et al., 2006; Roller and Erk, 2016; Roller et al., 2018). While there are effective patterns, such as X *is a* Y, that are indicative for entailment between nouns, there is little work on comparable patterns for verbs. Schwartz et al. (2015) mine symmetric patterns for lexical similarity and achieve good results for verbs. Entailment, however, is not symmetric.

Chklovski and Pantel (2004) handcrafted 35 patterns to distinguish 6 semantic relations for pairs of distributionally similar verbs. Some of their classes like strength (*taint :: poison*) or antonymy (*ban :: allow*) can be indicators of entailment and non-entailment but are, in general, much more narrowly defined than the patterns we use in our approach. Another difference to our work is that verb pairs are scored based on co-occurrence counts on the web, while we employ an LM, which does not depend on a valid entailment pair actually appearing together in a document.

Patterns and language models. Amrami and Goldberg (2018) were the first to manipulate LM predictions with a simple pattern to enhance the quality of substitute words in a given context for word sense induction. Petroni et al. (2019) found that large pretrained LMs can be queried for factual knowledge, when presented with appropriate pattern-generated cloze-style sentences. This zero-shot factual knowledge has later been shown to be quite fragile (Kassner and Schütze, 2020). So we rather focus on approaches that fine-tune an LM on at least a few samples. Forbes et al. (2019) train a binary classifier on top of a fine-tuned BERT (Devlin et al., 2019a) to predict the truth value of handwritten statements about objects and their properties. While their experiments investigate BERT's physical common sense reasoning, we focus on the different phenomenon of entailment between two actions expressed by verbs in context.

Schick and Schütze (2020) used handcrafted patterns and LMs for few-shot text classification. Based on manually defined label-token correspondences, the predicted classification label is determined by the token an LM estimates as most probable at a masked position in the cloze-style pattern. We differentiate entailment and non-entailment via compatibility scores for patterns and antipatterns and not via different predicted tokens.

Addressing relation induction, Bouraoui et al. (2020) propose an automatic way of finding, given a relation, LM patterns that are likely to express it. They train a binary classifier per relation on the sentences generated by these patterns. While some of the relations they consider are related to verbal entailment (e.g., *cook activity-goal eat*), most of them concern common sense (e.g., *library location-activity reading*) or encyclopedic knowledge (e.g., *Paris capital-of France*). We adapt their method for the automatic retrieval of promising patterns for LIiC, but find that handcrafted patterns that capture the generality of the entailment relation still have an advantage over automatic patterns for LIiC. Another important novelty we introduce is the use of antipatterns. While Bouraoui et al. (2020) have to use negative samples for training their classifiers, they only consider patterns that exemplify the desired relation. In contrast, we also use antipatterns that exemplify what the entailment relation is **not**. We believe that antipatterns are particularly useful for entailment detection because they can help identify other kinds of semantic relations that often pose a challenge to vector space models (Levy and Dagan, 2016; Schmitt and Schütze, 2019).

3 Proposed Approaches

3.1 NLI classifier

Building an NLI classifier on top of a pretrained LM usually means taking an aggregate sequence representation of the concatenated premise and hypothesis as input features of a neural network classifier (Devlin et al., 2019b). For RoBERTa (Liu et al., 2019), this representation is the final hidden state of a special $\langle \mathtt{s} \rangle$ token that is prepended to the input sentences, which in turn are separated by a separator token $\langle / \mathtt{s} \rangle$. Let Λ be the function that maps such an input $x = x_1 \langle / \mathtt{s} \rangle x_2$ to the aggregate representation $\Lambda(x) \in \mathbb{R}^d$. Following (Devlin et al., 2019b; Liu et al., 2019), we then feed these features to a 2-layer feed-forward neural network with tanh activation:

$$\begin{aligned} h(x) &= \tanh(\mathrm{drop}(\Lambda(x))W_1 + b_1) \\ P_{\text{NLI}}(y \mid x) &= \sigma(\mathrm{drop}(h(x))W_2 + b_2) \end{aligned} \tag{1}$$

where drop applies dropout with a probability of 0.1, σ is the softmax function, and $W_1 \in \mathbb{R}^{d\times d}, W_2 \in \mathbb{R}^{d\times 2}, b_1 \in \mathbb{R}^d, b_2 \in \mathbb{R}^2$ are learnable parameters. Note that W_1 and b_1 are still part of the LM's pretrained parameters; so we only train W_2 and b_2 from scratch.[2] The actual classification decision uses a threshold ϑ:

$$D^{\vartheta}_{\text{NLI}}(x_1, x_2) = \begin{cases} 1, & \text{if } P_{\text{NLI}}(y = 1 \mid x_1, x_2) > \vartheta \\ 0, & \text{otherwise} \end{cases}$$

The traditional choice for the threshold is $\vartheta = 0.5$ because that means $D^{\vartheta}_{\text{NLI}}(x_1, x_2) = 1$ iff $P_{\text{NLI}}(y = 1 \mid x_1, x_2) > P_{\text{NLI}}(y = 0 \mid x_1, x_2)$. We nevertheless keep ϑ as a hyperparameter to be tuned on held-out development data.

We train the NLI approach by minimizing the negative log-likelihood $\mathcal{L}_{\text{NLI}}$ of the training data $\mathcal{T}$:

$$\mathcal{L}_{\text{NLI}}(\mathcal{T}) = \sum_{(x_1,x_2,y)\in\mathcal{T}} -\log(P_{\text{NLI}}(y \mid x_1, x_2))$$

3.2 Pattern-based classifier

This approach puts the input sentences x_1, x_2 together in a pattern-based textual context and trains a classifier to distinguish between felicitous and infelicitous utterances.[3] In contrast to previous approaches (Forbes et al., 2019; Bouraoui et al., 2020), we also consider antipatterns that exemplify what kind of semantic relatedness we are not interested in, and combine probabilities for patterns and antipatterns in the final classification.

Finding suitable patterns. A simple handcrafted pattern to check for the validity of an inference $x_1 \Rightarrow x_2$ is "x_2 *because* x_1.". An analoguos antipattern is "*It is not sure that* x_2 *just because* x_1.". Based on similar considerations, we manually design 5 patterns and 5 antipatterns (see Table 4). We will refer to the approach using these handcrafted patterns as MANPAT.

Bouraoui et al. (2020) argue that text produced by simple, handcrafted patterns is artificial and

[2] We follow the official implementation; cf. Jacob Devlin's comment on issue 43 in the BERT GitHub repository, https://github.com/google-research/bert/issues/43, (accessed 19 January 2021).

[3] Bouraoui et al. (2020) called this natural vs. unusual.

therefore suboptimal for LMs pretrained on naturally occurring text. To adapt their setup to verbal expressions used in LIiC, we identify suitable patterns (antipatterns) by searching a large text corpus[4] for sentences that contain both elements of valid (invalid) entailment pairs. In a second step, we score each of these patterns (antipatterns) according to the number of valid (invalid) entailment pairs x_1, x_2 that can be found by querying an LM for the k most probable completions when x_1 or x_2 is inserted in the pattern and its counterpart is masked. For example, consider the entailment pair *rule* ⇒ *control* and the pattern "*Catchers* ***prem*** *the field; they* ***hypo*** *the plays and tell everyone where to be.*" extracted from a description of softball. Predicting *rule* from "*Catchers* ⟨`mask`⟩ *the field; they control the plays and tell everyone where to be.*" and predicting *control* from "*Catchers rule the field; they* ⟨`mask`⟩ *the plays and tell everyone where to be.*" would result in one point each. Approaches called AUTPAT$_n$ use the n patterns with the most points obtained in that manner. See §4 for more details on our experimental setup.

Pattern-based predictions. The probability $P_{\text{FEL}}(z \mid x)$ of sentence x to be felicitous ($z=1$) or infelicitous ($z=0$) is estimated like P_{NLI} in Eq. (1), except that x is not the concatenation of two sentences but a single pattern-generated utterance.

Given a set of patterns Φ and a set of antipatterns Ψ, the score s to judge an input x_1, x_2 is the difference between the maximum probability m_{pos} that any pattern forms a felicitous statement and the maximum probability m_{neg} that any antipattern forms a felicitous statement:

$$\begin{aligned} m_{\text{pos}} &= \max_{\varphi\in\Phi} P_{\text{FEL}}(z=1 \mid \varphi(x_1,x_2)) \\ m_{\text{neg}} &= \max_{\psi\in\Psi} P_{\text{FEL}}(z=1 \mid \psi(x_1,x_2)) \\ s(x_1,x_2) &= m_{\text{pos}} - m_{\text{neg}} \end{aligned}$$

As in NLI, the final decision uses a threshold ϑ:

$$D^{\vartheta}_{\text{PAT}}(x_1,x_2) = \begin{cases} 1, & \text{if } s(x_1,x_2) > \vartheta \\ 0, & \text{otherwise} \end{cases}$$

This corresponds to requiring that m_{pos} be higher than m_{neg} by a margin ϑ, i.e., $D^{\vartheta}_{\text{PAT}}(x_1,x_2) = 1$ iff $m_{\text{pos}} > m_{\text{neg}} + \vartheta$.

As Bouraoui et al. (2020) did not use antipatterns, they defined m_{neg} as the maximum probability for any pattern to form an infelicitous statement. To estimate the usefulness of antipatterns, we evaluate both possibilities, marking systems that use both patterns and antipatterns with $\Phi\Psi$ and those that only use patterns with Φ.

		Levy/Holt	SherLIiC
dev$_1$	train	4,388	797
	dev$_2$	1,098	201
test		12,921	2,990

Table 1: Data split sizes as used in our experiments.

The use of a threshold is another novel component, i.e., Bouraoui et al. (2020) virtually set $\vartheta = 0$. We discuss the influence of ϑ in §5.

We train all pattern-based approaches by minimizing the negative log-likelihood $\mathcal{L}_{\text{PAT}}$ that patterns Φ produce felicitous statements for valid entailments ($y=1$) and infelicitous statements for invalid entailments ($y=0$) from the training data $\mathcal{T}$, and vice versa for antipatterns Ψ:

$$\mathcal{L}_{\text{PAT}}(\mathcal{T},\Phi,\Psi) = \sum_{(x_1,x_2,y)\in\mathcal{T}} \mathcal{L}_{\Phi}(x_1,x_2,y) + \mathcal{L}_{\Psi}(x_1,x_2,1-y)$$

with

$$\mathcal{L}_{\Omega}(x_1,x_2,y) = -\frac{1}{|\Omega|}\sum_{\omega\in\Omega} \log(P_{\text{FEL}}(z=y \mid \omega(x_1,x_2)))$$

4 Experiments

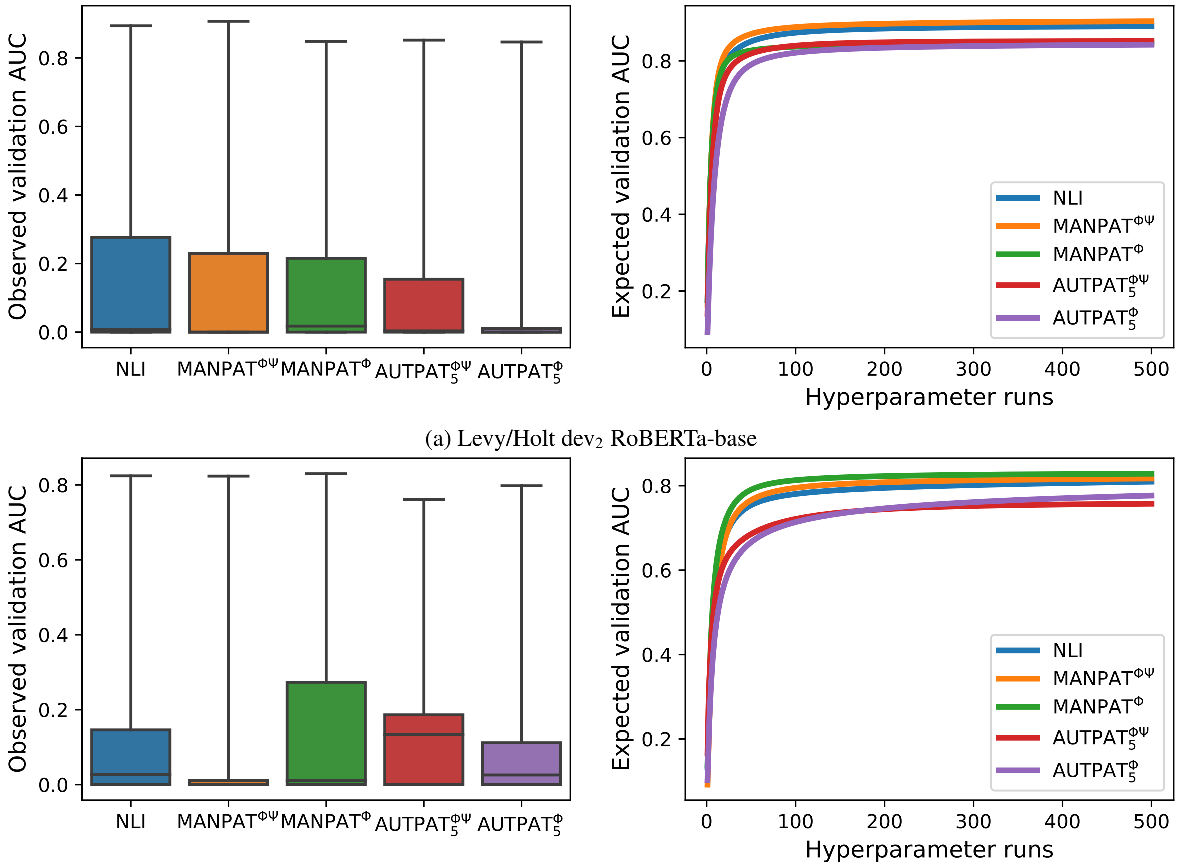
We evaluate on two benchmarks: (i) Levy/Holt's dataset (Levy and Dagan, 2016; Holt, 2018) and (ii) SherLIiC (Schmitt and Schütze, 2019). For both filtering and classification, we employ RoBERTa-base (Liu et al., 2019). For classification only, we also report results for RoBERTa-large.

4.1 Data processing

For both datasets, previous work has established a dev/test split. For Levy/Holt, it was defined in (Hosseini et al., 2018); for SherLIiC, we use the original one from (Schmitt and Schütze, 2019). For comparison with previous work, we keep the test portion as is and split the dev portion further into 80% for training and 20% for development. We call the new, smaller dev sets dev$_2$ and the original dev sets dev$_1$. See Table 1 for data split sizes.

Levy/Holt. An instance in Levy/Holt has two sentences, each consisting of two shared noun

[4] We use the Wikipedia dump from Jan 15th 2011.



(a) Levy/Holt dev$_2$ RoBERTa-base

(b) SherLIiC dev$_2$ RoBERTa-base

Figure 1: Validation performance distribution of different datasets across different hyperparameter runs (left) and expected validation performance per number of tested hyperparameter configurations as proposed by Dodge et al. (2019) (right). Performance is measured as the area under the precision-recall curve for precision values ≥ 0.5. The Boxes represent 75% of the respective data points; a black line indicates the median, whiskers extend to the maximum value.

phrases (the arguments) and a verbal expression, in which the two sentences differ. As the verbal expressions can contain auxiliaries or negation, they often consist of multiple tokens. Originally, one argument is replaced with a WordNet (Miller, 1995) type in one of the sentences to make the entailment more general during annotation, but we use a version of the dataset provided by Hosseini et al. (2018) where both sentences have concretely instantiated arguments. For example, consider Table 6 (c). *Athena* was masked as the WordNet synset *deity* during benchmark annotation but we use the original sentences as shown in Table 6 for all classifiers without further modification.

For the automatic pattern search in AUTPAT, we look for sentences that mention verbatim the two verbal expressions of any instance from dev$_1$. For the ranking, we take the last token of a verbal expression as representative for the whole. This has the advantage that we can query the LM with a single $\langle$mask$\rangle$ token and compare a single token to the $k = 100$ most probable predictions. We take the last token because it usually is the main verb.

SherLIiC. For classification, we use SherLIiC's automatically generated sentences that were used for annotation during benchmark creation. The arguments in SherLIiC are entity types from Freebase (Bollacker et al., 2008). As such, they can be replaced by any Freebase entity with matching type. For example, consider Table 6 (a); the arguments *Germany* and *Côte d'Ivoire* were originally masked as *location[A]* and *location[B]* during annotation, but annotators also saw three randomly chosen instantiations for both *A* (*Germany / Syria / USA*) and *B* (*Côte d'Ivoire / UK / Italy*) for context. From the three examples provided in SherLIiC for each argument, we choose the first one to form sentences with concretely instantiated arguments.

For the automatic pattern search in AUTPAT, we make use of the greater flexibility offered by the

		AUC	P	R	F1
baselines					
Hosseini et al. (2018)		16.5	–	–	–
Hosseini et al. (2019)		**18.7**	–	–	–
RoBERTa-base					
NLI	($\vartheta =$ 0.0052)	72.6	68.7	**75.3**	71.9
MANPAT$^{\Phi\Psi}$	($\vartheta = -0.0909$)	**76.9**	**78.7**	66.4	**72.0**
MANPAT$^{\Phi}$	($\vartheta =$ 0.5793)	71.2	74.4	61.2	67.1
AUTPAT$_5^{\Phi\Psi}$	($\vartheta = -0.1428$)	63.7	71.0	58.8	64.3
AUTPAT$_5^{\Phi}$	($\vartheta = -0.0592$)	65.4	68.0	63.3	65.5
RoBERTa-large					
NLI	($\vartheta =$ 0.0016)	75.5	73.5	73.7	73.6
MANPAT$^{\Phi\Psi}$	($\vartheta =$ 0.1156)	**83.9**	**84.8**	70.1	**76.7**
MANPAT$^{\Phi}$	($\vartheta = -0.8457$)	77.8	67.9	**81.5**	74.1
AUTPAT$_5^{\Phi\Psi}$	($\vartheta = -0.0021$)	70.4	75.7	60.7	67.4
AUTPAT$_5^{\Phi}$	($\vartheta = -0.9197$)	66.5	61.8	74.4	67.5

Table 2: Levy/Holt test. AUC denotes the area under the precision-recall curve for precision ≥ 0.5. All results in %. Bold means best result per column and block.

		AUC	P	R	F$_1$
baselines					
Lemma		–	**90.7**	8.9	16.1
w2v+untyped_rel		–	52.8	69.5	60.0
w2v+tsg_rel_emb		–	51.8	**72.7**	**60.5**
RoBERTa-base					
NLI	($\vartheta =$ 0.3878)	65.8	**67.0**	66.1	66.5
MANPAT$^{\Phi\Psi}$	($\vartheta = -0.3324$)	66.4	60.9	78.8	68.7
MANPAT$^{\Phi}$	($\vartheta = -0.4812$)	**69.2**	62.0	**81.2**	**70.3**
AUTPAT$_5^{\Phi\Psi}$	($\vartheta = -0.4694$)	67.4	61.8	75.6	68.0
AUTPAT$_5^{\Phi}$	($\vartheta = -0.7042$)	67.3	56.6	82.6	67.2
AUTCUR$_5^{\Phi}$	($\vartheta = -0.7524$)	**69.5**	56.3	**89.6**	**69.2**
AUTARG$_5^{\Phi}$	($\vartheta = -0.7461$)	65.2	**61.9**	75.6	68.1
RoBERTa-large					
NLI	($\vartheta =$ 0.0025)	68.3	60.5	**85.5**	70.9
MANPAT$^{\Phi\Psi}$	($\vartheta = -0.0956$)	**74.4**	**66.0**	80.8	**72.6**
MANPAT$^{\Phi}$	($\vartheta = -0.6641$)	64.6	58.1	79.0	67.0
AUTPAT$_5^{\Phi\Psi}$	($\vartheta = -0.9889$)	68.6	61.9	75.5	68.0
AUTPAT$_5^{\Phi}$	($\vartheta = -0.5355$)	56.8	61.5	66.1	63.7

Table 3: SherLIiC test. Baseline results from (Schmitt and Schütze, 2019). Table format: see Table 2.

lemmatized representations in SherLIiC. As we are interested in statements that can be made in any way in a text, we search for sentences that mention the two predicates of a SherLIiC dev$_1$ instance in any inflected form. For the ranking, we again consider the predicate representative for the whole verbal expression. We thus use the predicate lemma and otherwise proceed as described above.

4.2 Training details

We train all our classifiers for 5 epochs with Adam (Kingma and Ba, 2015) and a mini-batch size of 10 (resp. 2) for RoBERTa-base (resp. -large). We randomly sample 500 configurations for the remaining hyperparameters (see Appendix A). For a fair comparison, we evaluate all our approaches with the same configurations.

5 Results and Discussion

5.1 Hyperparameter robustness

Following previous work (Hosseini et al., 2018, 2019), we use the area under the precision-recall curve (AUC) restricted to precision values ≥ 0.5 as criterion for model selection.

Fig. 1 (left) shows the distribution of dev$_2$ performance for 500 randomly sampled runs with RoBERTa-base. Most hyperparameters perform poorly, suggesting that hyperparameter search is crucial. For Levy/Holt, NLI is strong whereas for SherLIiC handcrafted MANPAT$^{\Phi}$ patterns have a clearer advantage. For SherLIiC, the combination of automatically generated patterns and antipatterns AUTPAT$_5^{\Phi\Psi}$ exhibits the highest median performance and the second-highest upper quartile, making it together with MANPAT$^{\Phi}$ the most robust to different hyperparameters, although its top performance is lower compared to the others. For all methods, only very few hyperparameter sets achieve top performances. For both datasets, however, a well-performing configuration is found after fewer than 100 sampled runs (Fig. 1, right). Considering that AUTPAT requires an LM to rank thousands of patterns, these results suggest that, for LIiC, available GPU hours should be spent on automatic hyperparameter rather than pattern search. With its manually written patterns, MANPAT does not need additional GPU hours for pattern search and still, on average, performs better.

5.2 Best hyperparameter configurations

For the best found configuration for each method, we not only report AUC, which provides a general picture of a scoring method's precision-recall tradeoff, but also the concrete precision, recall, and F1 for the actual classification after applying a threshold ϑ. For this we tune ϑ on dev$_2$ for optimal F1. Tables 2 and 3 show the results.

On both datasets, our methods outperform all previous work (sometimes by a large margin), thus establishing a new state of the art. For SherLIiC+RoBERTa-base, the strong but simple NLI system is consistently outperformed by all pattern-based approaches, showing that well-

Automatically retrieved patterns (with SherLIiC dev$_1$)		prem	hypo
rank 1	In North America, where the ”atypical” forms of community-**hypo** pneumonia are becoming more common, macrolides (such as azithromycin), and doxycycline have displaced amoxicillin as first-line outpatient treatment for community-**prem** pneumonia.	acquired	acquired
rank 5	This area now consists of … the Yukon Territory (**prem** 1898) … and Nunavut (**hypo** 1999).	created	created in
rank 12	For example, … 訪問 ”**prem**” is composed of 訪 ”to visit” and 問 ”to **hypo**”.	interview	ask
Handcrafted patterns			
(a)	PARGL **prem** PARGR, which means that HARGL **hypo** HARGR.		
(b)	It is not the case that HARGL **hypo** HARGR, let alone that PARGL **prem** PARGR.		
(c)	HARGL **hypo** HARGR because PARGL **prem** PARGR.		
(d)	PARGL **prem-negated** PARGR because HARGL **hypo-negated** HARGR.		
(e)	HARGL **hypo-negated** HARGR, which means that PARGL **prem-negated** PARGR.		

Table 4: Examples of automatically retrieved and ranked AUTPAT patterns (top) and handcrafted MANPAT patterns (bottom). prem/hypo = original fillers as found in the corpus. PARGL/HARGL = placeholder for left argument of premise/hypothesis; PARGR/HARGR = right argument.

chosen patterns and antipatterns can be helpful for LIiC. For SherLIiC+RoBERTa-large and also generally on Levy/Holt's dataset, NLI is more competitive, but the combination of handcrafted patterns and antipatterns MANPAT$^{\Phi\Psi}$ still performs better in these cases.

The use of antipatterns does not consistently lead to better performance for all combinations of dataset, LM variant (base vs. large), and pattern set (MANPAT vs. AUTPAT). They do, however, consistently bring gains for some combinations, e.g., MANPAT on Levy/Holt and AUTPAT on SherLIiC. Moreover, antipatterns are essential for achieving top performance, i.e., the new state of the art, on both datasets.

Most of the threshold values ϑ (tuned on dev$_2$) are far from their traditional values, 0.5 for NLI and 0.0 for patterns. NLI classifiers' probability estimates are often too confident, resulting in values close to 0 and 1. To “correct” cases where a very small value is assigned to a valid entailment, optimal thresholds are often close to 0 instead of 0.5. Analogously, most pattern-based approaches opt for a negative ϑ, which means that instead of requiring a margin between m_{pos} and m_{neg} (boosting precision), they make more positive predictions and boost recall. Low recall is a key problem in LIiC (cf. Levy and Dagan (2016)). Tuning a threshold increases the models' flexibility in this aspect.

	$\Phi\Psi$		Φ	
n	AUC	F1	AUC	F1
5	67.4	68.0	67.3	67.2
15	**70.0**	**68.7**	**73.1**	**69.4**
25	63.5	67.3	69.0	68.7
50	66.3	65.6	67.4	67.6

Table 5: RoBERTa-base+AUTPAT$_n$ results on SherLIiC test for different n values. Hyperparameters were tuned for the corresponding AUTPAT$_5$ method on dev$_2$.

6 Analysis

6.1 Number of patterns

§5 shows that automatic patterns do not beat handcrafted patterns for LIiC. However, automatic patterns have one major advantage: in contrast to manual patterns, their number can be easily increased. We therefore investigate the impact of the hyperparameter n for AUTPAT$_n$.

Table 5 shows that too many patterns is as bad as too few. AUTPAT$_{15}$ is the sweetspot: on SherLIiC, it outperforms all other RoBERTa-base methods on AUC and closely approaches the otherwise best method MANPAT$^{\Phi}$ on F1.

6.2 Pattern analysis

Handcrafted patterns mostly outperform automatic ones (§5). A larger number n of patterns only has a small effect (§6.1). We therefore take a closer look at automatic and manual patterns. Table 4 shows all handcrafted and a sample of highly ranked automatic patterns.

It is striking how specific the automatically retrieved contexts are; especially for the highest ranks (exemplified by ranks 1 and 5) only a narrow set of verbs seems plausible from a human perspective. It is only at rank 12 that we find a more general context and it arguably even displays some semantic reasoning. There certainly are verbs that are not compatible with the meaning of *visit*, but this context allows for a wide range of plausible verbs and even mentions composition of meaning.

The handcrafted patterns, in contrast, all capture some general aspect of entailment, which might be the reason they generalize better. Moreover, they also have placeholder slots for the verb arguments, which could be an advantage as these represent a verb’s original context. Only accepting corpus sentences in which the verbs occur with the same arguments as in the dataset is too restrictive.

We therefore conduct the following experiment: We manually go through the 100 highest-ranked automatically created patterns and identify 5 contexts that could accommodate arguments without changing the overall sentence structure. We also try to pick patterns that are different enough from each other to avoid redundancy. As a baseline, the method AUTCUR$_5^{\Phi}$ uses these manually curated patterns as is. We then rewrite the patterns such that they include placeholders for verb arguments, e.g., “*The original aim of de Garis’ work was to* ***prem*** *the field of ”brain building” (a term of his invention) and to ”****hypo*** *a trillion dollar industry within 20 years”*.” becomes “*The original aim of their work was that ”*PARGL ***prem*** PARGR*” and that ”*HARGL ***hypo*** HARGR *within 20 years”*.” with PARGL / PARGR (HARGL / HARGR) the placeholder for the left / right argument of the premise (hypothesis). See Table 14 in the appendix for the complete list. AUTARG$_5^{\Phi}$ is based on these rewritten patterns. We try the same 500 hyperparameter configurations as for the other RoBERTa-base approaches and include results for the best configuration (chosen on dev$_2$) in Table 3. We find that manually curating automatically ranked patterns helps performance. AUTCUR$_5^{\Phi}$ outperforms AUTPAT$_5^{\Phi}$ on AUC and F1, reducing the gap to handcrafted patterns (i.e., MANPAT$^{\Phi}$). This is probably due to the variety we enforced when handpicking the patterns.

Surprisingly, adding arguments decreases performance. Possibly, our modifications make the patterns less fluent or the arguments that are filled into the placeholders during training and evaluation do not fit well into the contexts, which still are rather specific.

6.3 Error analysis

Table 6 displays a selection of the dev$_2$ sets of our two benchmarks along with the predictions of all our approaches.

(a) Sh	*Germany is occupying Côte d’Ivoire* ⇒ *Germany is remaining in Côte d’Ivoire* truth: 1 NLI: 0 MANPAT: 1 / 0 AUTPAT: 1 / 1
(b) L/H	*Ford awarded him the medal* ⇒ *Ford was awarded a medal* truth: 0 NLI: 1 MANPAT: 0 / 0 AUTPAT: 1 / 1
(c) L/H	*Athena was worshiped in Athens* ⇒ *Athena was the goddess of Athens* truth: 0 NLI: 0 MANPAT: 0 / 1 AUTPAT: 1 / 1
(d) L/H	*Pyrrhus was beaten by the romans* ⇒ *Pyrrhus fought the romans* truth: 1 NLI: 1 MANPAT: 0 / 0 AUTPAT: 0 / 1
(e) Sh	*England national rugby union team is playing against Denver Broncos* ⇒ *England national rugby union team is beating Denver Broncos* truth: 0 NLI: 1 MANPAT: 1 / 1 AUTPAT: 1 / 1
(f) L/H	*Polk negotiated with Britain* ⇒ *Polk made peace with Britain* truth: 0 NLI: 1 MANPAT: 1 / 1 AUTPAT: 1 / 1

Table 6: Qualitative error analysis of the RoBERTa-base models from Tables 2 and 3 on the dev$_2$ split of SherLIiC (Sh) and Levy/Holt (L/H). Pattern-based predictions are listed in the format $\Phi\Psi$ / Φ. Correct predictions are green; errors are underlined and red.

The first four examples indicate how NLI differs from pattern approaches. Example (a) involves the common sense knowledge that occupying a territory implies remaining there. This might be learned from patterns more easily as these patterns might resemble contexts – seen during pretraining – that describe how long a military force remained during an occupation. Putting the inference candidate (b) into a pattern-generated context avoids being fooled by the high similarity of the two sentences. Only the handcrafted patterns can make sense of the important details in this construction.

In contrast, (c) and (d) are difficult for our pattern approaches whereas NLI gets them right. We hypothesize that the problem stems from linking the two sentences into one. An entailment pattern ideally represents a derivation of the hypothesis from the premise. One may wrongly conclude that (c) *Athena was the goddess of Athens only because she was worshiped there*, by neglecting the possibility that there are others that are equally worshiped. In the same way, (d) is unlikely to be found in an argumentative text. While it is clear that there can be no beating without a fight, one would hardly argue that *Pyrrhus fought the romans because they beat him*. This particular reasoning calls for additional explanations like *Pyrrhus must have fought the romans because I know that they beat him*. This analysis serves as inspiration for further improvements of entailment patterns.

The last two examples (e) and (f) are difficult for all approaches. It seems to be a particular challenge to identify open situations like a sports match or a negotiation where multiple outcomes are possible and distinguish them from cases where one particular outcome is inevitable.

7 Conclusion

We proposed and evaluated three approaches to the task of lexical inference in context (LIiC) based on pretrained language models (LMs). In particular, we found that putting an inference candidate into a pattern-generated context mostly increases performance compared to a standard sequence classification approach. Concrete performance, however, also depends on the particular dataset, used LM (variant), and pattern set. We introduced the concept of antipatterns, which express the negative class of a binary classification, and found that they often lead to performance gains for LIiC. We set a new state of the art for LIiC and conducted an extensive analysis of our approaches. Notably, we found that automatically created patterns can perform nearly as well as handcrafted ones if we either use the right number n of patterns or manually identify the right subset of them. Promising directions for future work are the investigation of alternative automatic pattern generation methods or a better modeling of the remaining challenges we described in our error analysis (§6.3).

Acknowledgments

We gratefully acknowledge a Ph.D. scholarship awarded to the first author by the German Academic Scholarship Foundation (Studienstiftung des deutschen Volkes). This work was supported by the BMBF as part of the project MLWin (01IS18050).

References

Asaf Amrami and Yoav Goldberg. 2018. Word sense induction with neural biLM and symmetric patterns. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4860–4867, Brussels, Belgium. Association for Computational Linguistics.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619, Portland, Oregon, USA. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Computing Research Repository*, arXiv:2005.14165.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 33–40, Barcelona, Spain. Association for Computational Linguistics.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing textual entailment: Models and applications*. Morgan & Claypool Publishers.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding. *Computing Research Repository*, arXiv:1810.04805.

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

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194, Hong Kong, China. Association for Computational Linguistics.

Christiane Fellbaum and George A. Miller. 1990. Folk psychology or semantic entailment? comment on rips and conrad (1989). *Psychological Review*, 97(4):565–570.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense? In *Proceedings of the 41th Annual Meeting of the Cognitive Science Society (CogSci 2019)*, pages 1753–1759. cognitivesciencesociety.org.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*.

Xavier R. Holt. 2018. Probabilistic models of relational implication. Master's thesis, Macquarie University.

Mohammad Javad Hosseini, Nathanael Chambers, Siva Reddy, Xavier R. Holt, Shay B. Cohen, Mark Johnson, and Mark Steedman. 2018. Learning typed entailment graphs with global soft constraints. *Transactions of the Association for Computational Linguistics*, 6:703–717.

Mohammad Javad Hosseini, Shay B. Cohen, Mark Johnson, and Mark Steedman. 2019. Duality of link prediction and entailment graph induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4736–4746, Florence, Italy. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Douwe Kiela, Laura Rimell, Ivan Vulić, and Stephen Clark. 2015. Exploiting image generality for lexical entailment detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 119–124, Beijing, China. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(04):359–389.

Omer Levy and Ido Dagan. 2016. Annotating relation inference in context via question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 249–255, Berlin, Germany. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland. Association for Computational Linguistics.

Mike Lewis and Mark Steedman. 2013. Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 768–774, Montreal, Quebec, Canada. Association for Computational Linguistics.

Dekang Lin and Patrick Pantel. 2001. DIRT: Discovery of Inference Rules from Text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 323–328, New York, NY, USA. ACM Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Computing Research Repository*, arXiv:1907.11692.

Yehudit Meged, Avi Caciularu, Vered Shwartz, and Ido Dagan. 2020. Paraphrasing vs coreferring: Two sides of the same coin. *Computing Research Repository*, arXiv:2004.14979.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Computing Research Repository*, arXiv:1301.3781.

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

Shachar Mirkin, Ido Dagan, and Maayan Geffet. 2006. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 579–586, Sydney, Australia. Association for Computational Linguistics.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

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

Stephen Roller and Katrin Erk. 2016. Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2163–2172, Austin, Texas. Association for Computational Linguistics.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–363, Melbourne, Australia. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *Computing Research Repository*, arXiv:2001.07676.

Martin Schmitt and Hinrich Schütze. 2019. SherLIiC: A typed event-focused lexical inference benchmark for evaluating natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 902–914, Florence, Italy. Association for Computational Linguistics.

Stefan Schoenmackers, Jesse Davis, Oren Etzioni, and Daniel Weld. 2010. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1088–1098, Cambridge, MA. Association for Computational Linguistics.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 258–267, Beijing, China. Association for Computational Linguistics.

Vered Shwartz, Omer Levy, Ido Dagan, and Jacob Goldberger. 2015. Learning to exploit structured resources for lexical inference. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 175–184, Beijing, China. Association for Computational Linguistics.

Vered Shwartz, Gabriel Stanovsky, and Ido Dagan. 2017. Acquiring predicate paraphrases from news tweets. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 155–160, Vancouver, Canada. Association for Computational Linguistics.

Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 849–856, Manchester, UK. Coling 2008 Organizing Committee.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ivan Vulić and Nikola Mrkšić. 2018. Specialising word vectors for lexical entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1134–1145, New Orleans, Louisiana. Association for Computational Linguistics.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 81–88.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

A Hyperparameters

We train all our classifiers for 5 epochs with the Adam optimizer (Kingma and Ba, 2015) and a mini-batch size of 10 or 2 instances for RoBERTa-base and -large, respectively. For AUTPAT$_n$ approaches with $n > 5$, we distribute the available patterns and antipatterns into chunks of size 5 for training to save memory. During evaluation, the predictions are based on all the patterns and antipatterns.

We randomly sample 500 configurations for the remaining hyperparameters, i.e., initial learning rate lr, weight decay λ (L2 regularization), and the number of batches c the gradient is accumulated before each optimization step, which virtually increases the batch size by a factor of c. The hyperparameters are sampled from the following intervals: $\mathrm{lr} \in [10^{-8}, 5 \cdot 10^{-2}]$, $\lambda \in [10^{-5}, 10^{-1}]$, $c \in \{1, 2, \ldots, 10\}$. lr and λ are sampled uniformly in log-space. For a fair comparison, we use the same 500 random configurations for all of our approaches.

As usual for Transformer models, we apply a learning rate schedule: lr decreases linearly such that it reaches 0 at the end of the last epoch. We do not employ warm-up.

The best configurations can be seen in Tables 8 and 10 for Levy/Holt's dataset and in Tables 9 and 11 for SherLIiC.

B Results on development sets

See Tables 12 and 13.

C Varying n in training and evaluation

Another approach to make use of different values of n in AUTPAT$_n$ systems is to vary n from training to evaluation. Figure 2 is a visualization of the performance impact of this procedure. The base point for the visualization (in white) is the AUC performance of AUTPAT$_5^\Phi$. We see that training with $n = 50$ almost always leads to a performance drop (marked in blue) w.r.t. this number. It seems generally to be catastrophic to evaluate a model with patterns that were not seen during training, indicating that there is no generalization from seen patterns to unseen patterns even if they were chosen by the same method and can thus be expected to be – at least to some extent – similar. In general, this evaluation suggests that modifying n after the training always leads to a drop in performance.

D Transfer between Datasets

Table 7 shows results on the question how well a model trained on one dataset performs on the other. For this, we assume that the target dataset is not available at all, i.e., we do not use it at all – neither for finding patterns in AUTPAT nor for tuning the threshold ϑ. We thus use the standard ϑ values, i.e., 0.5 for NLI and 0.0 for the pattern-based methods.

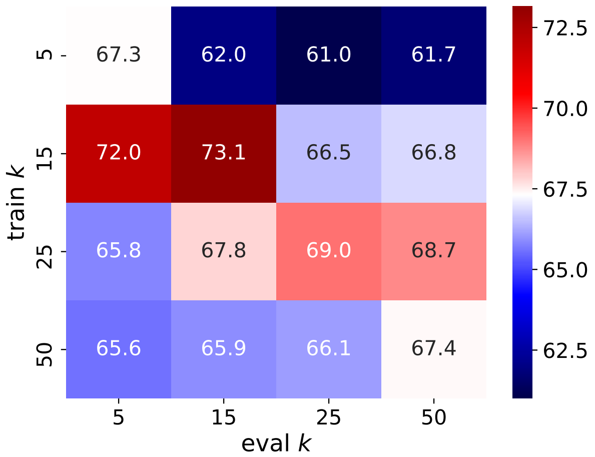


Figure 2: RoBERTa-base+AUTPAT$_k^\Phi$ performance on SherLIiC test for different k values during training and evaluation. Same hyperparameters used for all models (as in Table 5). Blue marks drops, red marks gains in performance w.r.t. AUTPAT$_5^\Phi$.

	AUC	P	R	F$_1$
RoBERTa-base				
NLI	38.4	52.7	57.1	**54.8**
MANPAT$^{\Phi\Psi}$	**46.1**	**64.0**	45.4	53.1
MANPAT$^{\Phi}$	32.4	32.4	**94.5**	48.2
AUTPAT$_5^{\Phi\Psi}$	18.7	40.5	35.0	37.6
AUTPAT$_5^{\Phi}$	21.3	28.3	62.3	38.9
RoBERTa-large				
NLI	37.8	31.0	96.4	46.9
MANPAT$^{\Phi\Psi}$	**70.4**	39.6	95.3	**56.0**
MANPAT$^{\Phi}$	38.9	25.6	**98.3**	40.6
AUTPAT$_5^{\Phi\Psi}$	33.6	**61.6**	36.0	45.5
AUTPAT$_5^{\Phi}$	9.3	30.7	76.6	43.8

(a) SherLIiC train → Levy/Holt test.

	AUC	P	R	F$_1$
RoBERTa-base				
NLI	63.3	62.8	**68.4**	**65.5**
MANPAT$^{\Phi\Psi}$	**69.1**	**80.5**	42.1	55.3
MANPAT$^{\Phi}$	68.4	80.1	24.2	37.2
AUTPAT$_5^{\Phi\Psi}$	60.3	71.5	54.5	61.9
AUTPAT$_5^{\Phi}$	58.9	68.6	55.7	61.5
RoBERTa-large				
NLI	65.6	73.8	53.0	61.7
MANPAT$^{\Phi\Psi}$	69.6	84.7	35.7	50.3
MANPAT$^{\Phi}$	**72.2**	**89.3**	30.3	45.2
AUTPAT$_5^{\Phi\Psi}$	62.1	68.1	**57.3**	**62.3**
AUTPAT$_5^{\Phi}$	63.8	75.8	44.2	55.8

(b) Levy/Holt train → SherLIiC test.

Table 7: Transfer learning. Table format: see Table 2.

	NLI	MANPAT$^{\Phi\Psi}$	MANPAT$^{\Phi}$	AUTPAT$_5^{\Phi\Psi}$	AUTPAT$_5^{\Phi}$
lr	$2.72 \cdot 10^{-5}$	$2.47 \cdot 10^{-5}$	$6.68 \cdot 10^{-6}$	$3.82 \cdot 10^{-5}$	$2.11 \cdot 10^{-5}$
λ	$1.43 \cdot 10^{-3}$	$2.98 \cdot 10^{-4}$	$1.07 \cdot 10^{-5}$	$4.02 \cdot 10^{-5}$	$1.65 \cdot 10^{-5}$
c	1	2	1	2	3

Table 8: Levy/Holt; RoBERTa-base.

	NLI	MANPAT$^{\Phi\Psi}$	MANPAT$^{\Phi}$	AUTPAT$_5^{\Phi\Psi}$	AUTPAT$_5^{\Phi}$	AUTCUR$_5^{\Phi}$	AUTARG$_5^{\Phi}$
lr	$6.34 \cdot 10^{-6}$	$3.87 \cdot 10^{-5}$	$2.28 \cdot 10^{-5}$	$3.92 \cdot 10^{-5}$	$2.53 \cdot 10^{-5}$	$1.28 \cdot 10^{-5}$	$2.47 \cdot 10^{-5}$
λ	$1.35 \cdot 10^{-3}$	$1.43 \cdot 10^{-5}$	$6.52 \cdot 10^{-2}$	$2.18 \cdot 10^{-4}$	$1.02 \cdot 10^{-5}$	$8.23 \cdot 10^{-3}$	$2.98 \cdot 10^{-4}$
c	1	4	2	1	1	1	2

Table 9: SherLIiC; RoBERTa-base.

	NLI	MANPAT$^{\Phi\Psi}$	MANPAT$^{\Phi}$	AUTPAT$_5^{\Phi\Psi}$	AUTPAT$_5^{\Phi}$
lr	$6.68 \cdot 10^{-6}$	$4.55 \cdot 10^{-6}$	$4.92 \cdot 10^{-6}$	$6.68 \cdot 10^{-6}$	$8.13 \cdot 10^{-6}$
λ	$1.07 \cdot 10^{-5}$	$3.90 \cdot 10^{-4}$	$3.61 \cdot 10^{-4}$	$1.07 \cdot 10^{-5}$	$6.05 \cdot 10^{-2}$
c	1	2	3	1	2

Table 10: Levy/Holt; RoBERTa-large.

	NLI	MANPAT$^{\Phi\Psi}$	MANPAT$^{\Phi}$	AUTPAT$_5^{\Phi\Psi}$	AUTPAT$_5^{\Phi}$
lr	$6.68 \cdot 10^{-6}$	$1.29 \cdot 10^{-5}$	$9.14 \cdot 10^{-6}$	$6.34 \cdot 10^{-6}$	$4.55 \cdot 10^{-6}$
λ	$1.07 \cdot 10^{-5}$	$2.49 \cdot 10^{-4}$	$6.09 \cdot 10^{-5}$	$1.35 \cdot 10^{-3}$	$3.90 \cdot 10^{-4}$
c	1	3	4	1	2

Table 11: SherLIiC; RoBERTa-large.

	dev$_1$				dev$_2$				test			
	AUC	P	R	F1	AUC	P	R	F1	AUC	P	R	F1
baselines												
Hosseini et al. (2018)	–	–	–	–	–	–	–	–	16.5	–	–	–
Hosseini et al. (2019)	–	–	–	–	–	–	–	–	**18.7**	–	–	–
RoBERTa-base												
NLI ($\vartheta = 0.0052$)	94.9	87.4	91.1	89.2	88.8	78.1	**90.3**	83.8	72.6	68.7	**75.3**	71.9
MANPAT$^{\Phi\Psi}$ ($\vartheta = -0.0909$)	**96.5**	**87.7**	**96.2**	**91.8**	**89.4**	**81.4**	88.5	**84.8**	**76.9**	**78.7**	66.4	**72.0**
MANPAT$^{\Phi}$ ($\vartheta = 0.5793$)	91.8	80.2	90.1	84.9	84.7	77.5	81.1	79.3	71.2	74.4	61.2	67.1
AUTPAT$_5^{\Phi\Psi}$ ($\vartheta = -0.1428$)	95.0	83.4	95.6	89.1	87.7	79.2	85.7	82.3	63.7	71.0	58.8	64.3
AUTPAT$_5^{\Phi}$ ($\vartheta = -0.0592$)	87.4	78.0	90.0	83.6	83.3	76.3	81.6	78.8	65.4	68.0	63.3	65.5
RoBERTa-large												
NLI ($\vartheta = 0.0016$)	96.9	90.1	97.1	**93.5**	87.7	82.6	87.6	**85.0**	75.5	73.5	73.7	73.6
MANPAT$^{\Phi\Psi}$ ($\vartheta = 0.1156$)	**97.1**	**91.4**	95.4	93.4	**88.9**	**84.0**	84.8	84.4	**83.9**	**84.8**	70.1	**76.7**
MANPAT$^{\Phi}$ ($\vartheta = -0.8457$)	92.2	76.1	**97.3**	85.4	84.4	72.5	**91.2**	80.8	77.8	67.9	**81.5**	74.1
AUTPAT$_5^{\Phi\Psi}$ ($\vartheta = -0.0021$)	95.0	86.0	91.9	88.8	84.7	78.9	81.1	80.0	70.4	75.7	60.7	67.4
AUTPAT$_5^{\Phi}$ ($\vartheta = -0.9197$)	92.4	75.5	95.3	84.2	83.5	70.6	88.5	78.5	66.5	61.8	74.4	67.5

Table 12: Full results on the Levy/Holt dataset. The dev and test sets as created by Hosseini et al. (2018) are called dev$_1$ and test. The portion of dev$_1$ that serves as our validation set is called dev$_2$. AUC denotes the area under the precision-recall curve for precision values ≥ 0.5. All results in %.

		dev$_1$				dev$_2$				test			
		AUC	P	R	F$_1$	AUC	P	R	F$_1$	AUC	P	R	F$_1$
baselines													
Lemma		–	**90.0**	10.9	19.4	–	–	–	–	–	**90.7**	8.9	16.1
w2v+untyped_rel		–	56.5	74.0	64.1	–	–	–	–	–	52.8	69.5	60.0
w2v+tsg_rel_emb		–	56.6	**77.6**	**65.5**	–	–	–	–	–	51.8	**72.7**	**60.5**
RoBERTa-base													
NLI	($\vartheta =$ 0.3878)	81.3	**79.1**	80.1	79.6	81.5	**84.2**	70.6	**76.8**	65.8	**67.0**	66.1	66.5
MANPAT$^{\Phi\Psi}$	($\vartheta = -0.3324$)	76.2	68.6	85.8	76.2	**82.4**	70.0	**82.4**	75.7	66.4	60.9	78.8	68.7
MANPAT$^{\Phi}$	($\vartheta = -0.4812$)	**88.4**	75.5	**93.1**	**83.4**	84.1	73.0	79.4	76.1	**69.2**	62.0	**81.2**	**70.3**
AUTPAT$_5^{\Phi\Psi}$	($\vartheta = -0.4694$)	87.0	77.8	88.8	82.9	71.2	68.4	76.5	72.2	67.4	61.8	75.6	68.0
AUTPAT$_5^{\Phi}$	($\vartheta = -0.7042$)	86.8	64.1	91.8	75.5	74.0	65.5	83.8	73.6	67.3	56.6	82.6	67.2
AUTCUR$_5^{\Phi}$	($\vartheta = -0.7524$)	82.6	61.7	92.8	74.1	75.6	60.6	88.2	71.9	69.5	56.3	89.6	69.2
AUTARG$_5^{\Phi}$	($\vartheta = -0.7461$)	77.4	69.3	84.0	76.0	73.6	68.9	75.0	71.8	65.2	61.9	75.6	68.1
AUTPAT$_{15}^{\Phi\Psi}$	($\vartheta = -0.5263$)	95.3	87.0	93.1	89.9	73.0	65.4	75.0	69.9	70.0	60.4	79.7	68.7
AUTPAT$_{15}^{\Phi}$	($\vartheta = -0.6422$)	**95.4**	85.3	94.6	89.7	**75.8**	69.2	**79.4**	**74.0**	**73.1**	**63.0**	77.4	**69.4**
AUTPAT$_{25}^{\Phi\Psi}$	($\vartheta = -0.0014$)	95.0	**92.0**	93.7	**92.8**	66.1	**70.0**	72.1	71.0	63.5	62.1	73.4	67.3
AUTPAT$_{25}^{\Phi}$	($\vartheta = -0.6496$)	88.1	72.3	90.0	80.2	73.0	67.5	79.4	73.0	69.0	60.5	**79.4**	68.7
AUTPAT$_{50}^{\Phi\Psi}$	($\vartheta = -0.9163$)	93.2	72.8	92.8	81.5	67.1	63.0	75.0	68.5	66.3	54.3	**82.8**	65.6
AUTPAT$_{50}^{\Phi}$	($\vartheta = -0.9500$)	94.2	79.1	**94.9**	86.3	69.3	66.3	77.9	71.6	67.4	57.3	82.5	67.6
RoBERTa-large													
NLI	($\vartheta =$ 0.0025)	**92.3**	**79.7**	**93.7**	**86.1**	75.7	66.7	**82.4**	73.7	68.3	60.5	**85.5**	70.9
MANPAT$^{\Phi\Psi}$	($\vartheta = -0.0956$)	89.3	77.3	88.5	82.5	**80.8**	**74.7**	77.9	**76.3**	**74.4**	**66.0**	80.8	**72.6**
MANPAT$^{\Phi}$	($\vartheta = -0.6641$)	78.0	67.4	84.9	75.1	72.2	67.1	77.9	72.1	64.6	58.1	79.0	67.0
AUTPAT$_5^{\Phi\Psi}$	($\vartheta = -0.9889$)	86.5	73.8	86.7	79.7	73.6	70.4	73.5	71.9	68.6	61.9	75.5	68.0
AUTPAT$_5^{\Phi}$	($\vartheta = -0.5355$)	71.6	64.3	71.9	67.9	64.5	71.4	66.2	68.7	56.8	61.5	66.1	63.7

Table 13: Full results on SherLIiC. The original dev and test sets are called dev$_1$ and test. The portion of dev$_1$ that serves as our validation set is called dev$_2$. AUC denotes the area under the precision-recall curve for precision values ≥ 0.5. Baseline results from (Schmitt and Schütze, 2019). All results in %.

(1)	The original aim of de Garis' work was to **prem** the field of "brain building" (a term of his invention) and to "**hypo** a trillion dollar industry within 20 years".	→	The original aim of their work was that "PARGL **prem** PARGR" and that "HARGL **hypo** HARGR within 20 years".
(2)	Critic Roger Ebert stated that Gellar and co-star Ryan Phillippe "**prem** a convincing emotional charge" and that Gellar is "effective as a bright girl who knows exactly how to **hypo** her act as a tramp".	→	Critic Roger Ebert stated that PARGL and co-star Ryan Phillippe "**prem** PARGR" and that HARGL is "effective as a bright girl who knows exactly how she **hypo** HARGR as a tramp".
(3)	Well-known professional competitions in the past have included the World Professional Championships (**hypo** Landover, Maryland), the Challenge Of Champions, the Canadian Professional Championships and the World Professional Championships (**prem** in Jaca, Spain).	→	Well-known professional competitions in the past have included HARGL (**hypo** HARGR), the Challenge Of Champions, the Canadian Professional Championships and PARGL (**prem** PARGR).
(4)	They also had sharpshooter Steve Kerr, whom they **hypo** via free agency before the 1993–94 season, Myers, and centers Luc Longley (**prem** via trade in 1994 from the Minnesota Timberwolves) and Bill Wennington.	→	HARGL also had sharpshooter HARGR, whom they **hypo** via free agency before the 1993–94 season, Myers, and centers PARGR (whom PARGL **prem** via trade in 1994 from the Minnesota Timberwolves) and Bill Wennington.
(5)	Because the 6x86 was more efficient on an instructions-per-cycle basis than Intel's Pentium, and because Cyrix sometimes **hypo** a faster bus speed than either Intel or AMD, Cyrix and competitor AMD co-**prem** the controversial PR system in an effort to compare its products more favorably with Intel's. ...	→	Because the 6x86 was more efficient on an instructions-per-cycle basis than Intel's Pentium, and because HARGL sometimes **hypo** HARGR, PARGL and competitor AMD co-**prem** PARGR in an effort to compare its products more favorably with Intel's. ...

Table 14: Five manually selected patterns from the 100 highest-ranked automatically extracted patterns from SherLIiC dev$_1$ (used in AUTCUR$_5^{\Phi}$) and their rewritten counterparts (used in AUTARG$_5^{\Phi}$). PARGL (HARGL) stands for the left argument of the premise (hypothesis); PARGR (HARGR) for the right one.