Ranking and Selecting Multi-Hop Knowledge Paths to Better Predict Human Needs

Debjit Paul
Research Training Group AIPHES
Institute for Computational Linguistics
Heidelberg University
paul@cl.uni-heidelberg.de

Anette Frank
Research Training Group AIPHES
Institute for Computational Linguistics
Heidelberg University
frank@cl.uni-heidelberg.de

Abstract

To make machines better understand sentiments, research needs to move from polarity identification to understanding the reasons that underlie the expression of sentiment. Categorizing the goals or needs of humans is one way to explain the expression of sentiment in text. Humans are good at understanding situations described in natural language and can easily connect them to the character's psychological needs using commonsense knowledge. We present a novel method to extract, rank, filter and select multi-hop relation paths from a commonsense knowledge resource to interpret the expression of sentiment in terms of their underlying human needs. We efficiently integrate the acquired knowledge paths in a neural model that interfaces context representations with knowledge using a gated attention mechanism. We assess the model's performance on a recently published dataset for categorizing human needs. Selectively integrating knowledge paths boosts performance and establishes a new state-of-the-art. Our model offers interpretability through the learned attention map over commonsense knowledge paths. Human evaluation highlights the relevance of the encoded knowledge.

1 Introduction

Sentiment analysis and emotion detection are essential tasks in human-computer interaction. Due to its broad practical applications, there has been rapid growth in the field of sentiment analysis (Zhang et al., 2018). Although state-of-the-art sentiment analysis can detect the polarity of text units (Hamilton et al., 2016; Socher et al., 2013), there has been limited work towards explaining the reasons for the expression of sentiment and emotions in texts (Li and Hovy, 2017). In our work, we aim to go beyond the detection of sentiment, toward explaining sentiments. Such explanations

can range from detecting overtly expressed explanations or reasons for sentiments towards specific aspects of, e.g., products or films, as in user reviews to the explanation of the underlying reasons for emotional reactions of characters in a narrative story. The latter requires understanding of stories and modeling the mental state of characters. Recently, Ding and Riloff (2018) proposed to categorize affective events with categories based on human needs, to provide explanations of people's attitudes towards such events. Given an expression such as *I broke my leg*, they categorize the reason for the expressed negative sentiment as being related to a need concerning 'health'.

In this paper we focus on the *Modelling Naive Psychology of Characters in Simple Commonsense Stories* dataset of Rashkin et al. (2018), which contains annotations of a fully-specified chain of motivations and emotional reactions of characters for a collection of narrative stories. The stories are annotated with labels from multiple theories of psychology (Reiss, 2004; Maslow, 1943; Plutchik, 1980) to provide explanations for the emotional reactions of characters.

Similar to Ding and Riloff (2018), we hypothesize that emotional reactions (joy, trust, fear, etc.) of characters can be explained by (dis)satisfaction of their psychological needs. However, predicting categories of human needs that underlie the expression of sentiment is a difficult task for a computational model. It requires not only detecting surface patterns from the text, but also requires commonsense knowledge about how a given situation may or may not satisfy specific human needs of a character. Such knowledge can be diverse and complex, and will typically be implicit in the text. In contrast, human readers can make use of relevant information from the story and associate it with their knowledge about human interaction, desires and human needs, and thus will be able

3671

to infer underlying reasons for emotions indicated in the text. In this work, we propose a computational model that aims to categorize human needs of story characters by integrating commonsense knowledge from ConceptNet (Speer and Havasi, 2012). Our model aims to imitate human understanding of a story, by (i) learning to select relevant words from the text, (ii) extracting pieces of knowledge from the commonsense inventory and (iii) associating them with human need categories put forth by psychological theories. Our assumption is that by integrating commonsense knowledge in our model we will be able to overcome the lack of textual evidence in establishing relations between expressed emotions in specific situations and the inferable human needs of story characters. In order to provide such missing associations, we leverage the graph structure of the knowledge source. Since these connections can be diverse and complex, we develop a novel approach to extract and rank multi-hop relation paths from ConceptNet using graph-based methods.

Our contributions are: (i) We propose a novel approach to extract and rank multi-hop relation paths from a commonsense knowledge resource using graph-based features and algorithms. (ii) We present an end-to-end model enhanced with attention and a gated knowledge integration component to predict human needs in a given context. To the best of our knowledge, our model is the first to advance commonsense knowledge for this task. (iii) We conduct experiments that demonstrate the effectiveness of the extracted knowledge paths and show significant performance improvements over the prior state-of-the-art. (iv) Our model provides interpretability in two ways: by selecting relevant words from the input text and by choosing relevant knowledge paths from the imported knowledge. In both cases, the degree of relevance is indicated via an attention map. (v) A small-scale human evaluation demonstrates that the extracted multi-hop knowledge paths are indeed relevant. Our code is made publicly available.[1]

2 Related Work

Sentiment Analysis and Beyond. Starting with Pang et al. (2002), sentiment analysis and emotion detection has grown to a wide research field. Researchers have investigated polarity classifica-

tion, sentiment and emotion detection and classification (Tang et al., 2015; Yin et al., 2017; Li et al., 2017) on various levels (tokens, phrases, sentences or documents), as well as structured prediction tasks such as the identification of holders and targets (Deng and Wiebe, 2015) or sentiment inference (Choi et al., 2016). Our work goes beyond the analysis of overtly expressed sentiment and aims at identifying goals, desires or needs underlying the expression of sentiment. Li and Hovy (2017) argued that the goals of an opinion holder can be categorized by human needs. There has been work related to goals, desires, wish detection (Goldberg et al., 2009; Rahimtoroghi et al., 2017). Most recently, Ding and Riloff (2018) propose to categorize affective events into physiological needs to explain people's motivations and desires. Rashkin et al. (2018) published a dataset for tracking emotional reactions and motivations of characters in stories. In this work, we use this dataset to develop a knowledge-enhanced system that 'explains' sentiment in terms of human needs.

Integrating Structured Knowledge into Neural NLU Systems. Neural models aimed at solving NLU tasks have been shown to profit from the integration of knowledge, using different methods: Xu et al. (2017) show that injecting loosely structured knowledge with a recall-gate mechanism is beneficial for conversation modeling; Mihaylov and Frank (2018) and Weissenborn et al. (2017) propose integration of commonsense knowledge for reading comprehension: the former explicitly encode selected triples from ConceptNet using attention mechanisms, the latter enriches question and context embeddings by encoding triples as mapped statements extracted from ConceptNet. Concurrently to our work, Bauer et al. (2018) proposed a heuristic method to extract multi-hop paths from ConceptNet for a reading comprehension task. They construct paths starting from concepts appearing in the question to concepts appearing in the context, aiming to emulate multi-hop reasoning. Tamilselvam et al. (2017) use ConceptNet relations for aspect-based sentiment analysis. Similar to our approach, Bordes et al. (2014) make use of knowledge bases to obtain longer paths connecting entities appearing in questions to answers in a QA task. They also provide a richer representation of answers by building subgraphs of entities appearing in answers. In contrast, our work aims to provide information about missing links
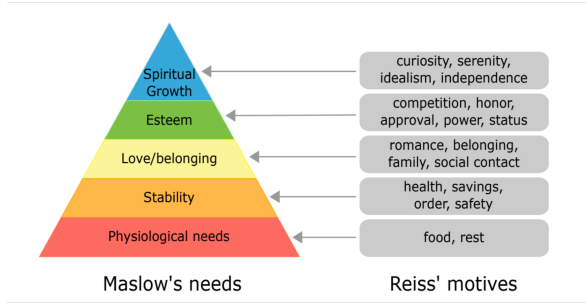

Figure 1: Maslow and Reiss: Theories of Psychology as presented in Rashkin et al. (2018).

Figure 2: Attention over multi-hop knowledge paths.

between sentiment words in a text and underlying human needs by extracting relevant multi-hop paths from structured knowledge bases.

3 Selecting and Ranking Commonsense Knowledge to Predict Human Needs

Our task is to automatically predict human needs of story characters given a story context. In this task, following the setup of Rashkin et al. (2018), we *explain* the probable reasons for the expression of emotions by predicting appropriate categories from two theories of psychology: *Hierarchy of needs* (Maslow, 1943) and *basic motives* (Reiss, 2002). The task is defined as a multi-label classification problem with five coarse-grained (Maslow) and 19 fine-grained (Reiss) categories, respectively (see Fig. 1).[1] We start with a Bi-LSTM encoder with self-attention as a baseline model, to efficiently categorize human needs. We then show how to select and rank multi-hop commonsense knowledge paths from ConceptNet that connect textual expressions with human need categories. Finally, we extend our model with a gated knowledge integration mechanism to incorporate relevant multi-hop commonsense knowledge paths for predicting human needs. An overview of the model is given in Figure 2. We now describe each component in detail.

3.1 A Bi-LSTM Encoder with Attention to Predict Human Needs

Our Bi-LSTM encoder takes as input a sentence S consisting of a sequence of tokens, denoted as $w_1^s, w_2^s, ..., w_n^s$, or $w_{1:n}^s$ and its preceding context Cxt, denoted as $w_1^{cxt}, w_2^{cxt}, ..., w_m^{cxt}$, or $w_{1:m}^{cxt}$. As further input we read the name of a story character, which is concatenated to the input sentence. For

---

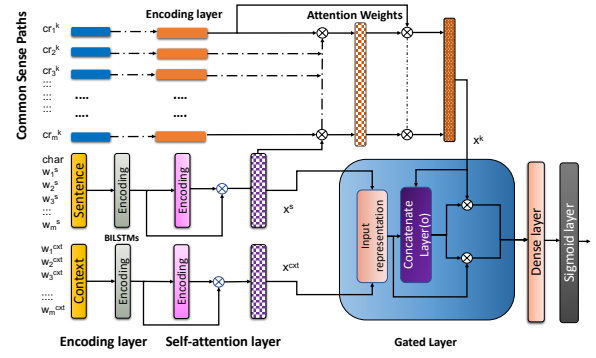[1]Details about the labels are given in the Supplement.

this input the model is tasked to predict appropriate human need category labels $z \in Z$, according to a predefined inventory.

Embedding Layer: We embed each word from the sentence and the context with a contextualized word representation using character-based word representations (ELMo) (Peters et al., 2018). The embedding of each word w_i in the sentence and context is represented as e_i^s and e_i^{cxt}, respectively.

Encoding Layer: We use a single-layer Bi-LSTM (Hochreiter and Schmidhuber, 1997) to obtain sentence and context representations h^s and h^{cxt}, which we form by concatenating the final states of the forward and backward encoders.

$$h^s = BiLSTM(e_{1:n}^s); h^{cxt} = BiLSTM(e_{1:m}^{cxt})$$
(1)

A Self-Attention Layer allows the model to dynamically control how much each token contributes to the sentence and context representation. We use a modified version of self-attention proposed by Rei and Søgaard (2018), where both input representations are passed through a feedforward layer to generate scalar values for each word in context v_i^{cxt} and sentence v_i^s (cf. (2-5)).

$$a_i^s = ReLU(W_i^s h_i^s + b_i^s),$$
(2)

$$a_i^{cxt} = ReLU(W_i^{cxt} h_i^{cxt} + b_i^{cxt})$$
(3)

$$v_i^s = W_v^s a_i^s + b_v^s i$$
(4)

$$v_i^{cxt} = W_v^{cxt} i a_i^{cxt} + b_v^{cxt} i$$
(5)

where, $W^s, b^s, W^{cxt}, b^{cxt}, W_v^s, W_v^{cxt}$ are trainable parameters. We calculate the soft attention weights for both sentence and context:

$$\widetilde{v}_i = \frac{1}{1 + exp(-v_i)}; \quad \hat{v}_i = \frac{\widetilde{v}_i}{\sum_{k=1}^N \widetilde{v}_k}$$
(6)

where, $\widetilde{v}_i$ is the output of the sigmoid function, therefore $\widetilde{v}_i$ is in the range [0,1] and $\hat{v}_i$ is the normalized version of $\widetilde{v}_i$. Values $\hat{v}_i$ are used as attention weights to obtain the final sentence and context representations x^s and x^{cxt}, respectively:

$$x^s = \sum_{i=1}^{N} \hat{v}_i^{\ s} h_i^s \quad (7) \qquad x^{cxt} = \sum_{i=1}^{M} \hat{v}_i^{\ cxt} h_i^{cxt} \tag{8}$$

with N and M the number of tokens in S and Cxt. The output of the self-attention layer is generated by concatenating x^s and x^{cxt}. We pass this representation through a FF layer of dimension Z:

$$y = ReLU(W_y[x^s; x^{cxt}] + b_y) \tag{9}$$

where W_y, b_y are trainable parameters and ';' denotes concatenation of two vectors. Finally, we feed the output layer y to a logistic regression layer to predict a binary label for each class $z \in Z$, where Z is the set of category labels for a particular psychological theory (Maslow/Reiss, Fig. 1).

3.2 Extracting Commonsense Knowledge

To improve the prediction capacity of our model, we aim to leverage external commonsense knowledge that connects expressions from the sentence and context to human need categories. For this purpose we extract multi-hop commonsense knowledge paths that connect words in the textual inputs with the offered human need categories, using as resource ConceptNet (Speer and Havasi, 2012), a large commonsense knowledge inventory. Identifying contextually relevant information from such a large knowledge base is a non-trivial task. We propose an effective two-step method to extract multi-hop knowledge paths that associate concepts from the text with human need categories: (i) collect all potentially relevant knowledge relations among concepts and human needs in a subgraph for each input sentence; (ii) rank, filter and select high-quality paths using graph-based local measures and graph centrality algorithms.

3.2.1 Construction of Sub-graphs

ConceptNet is a graph $G = (V, E)$ whose nodes are concepts and edges are relations between concepts (e.g. CAUSES, MOTIVATEDBY). For each sentence S we induce a subgraph $G' = (V', E')$ where V' comprises all concepts $c \in V$ that appear in S and the directly preceding sentence in

context Cxt. V' also includes all concepts $c \in V$ that correspond to one of the human need categories in our label set Z. Fig. 3 shows an example. The sub-graph is constructed as follows:

Shortest Paths: In a first step, we find all shortest paths p' from ConceptNet that connect any concept $c_i \in V'$ to any other concept $c_j \in V'$ and to each human needs concept $z \in Z$. We further include in V' all the concepts $c \in V$ which are contained in the above shortest paths p'.

Neighbours: To better represent the meaning of the concepts in V', we further include in V' all concepts $c \in V$ that are directly connected to any $c \in V'$ that is not already included in V'.

Sub-graph: We finally construct a connected sub-graph $G' = (V', E')$ from V' by defining E' as the set of all ConceptNet edges $e \in E$ that directly connect any pair of concepts $(c_i, c_j) \in V'$.

Overall, we obtain a sub-graph that contains relations and concepts which are supposed to be useful to "explain" *why* and *how strongly* concepts c_i that appear in the sentence and context are associated with any of the human needs $z \in Z$.

3.2.2 Ranking and Selecting Multi-hop Paths

We could use all possible paths p contained in the sub-graph G', connecting concepts c_i from the text and human needs concepts z contained in G', as additional evidence to predict suitable human need categories. But not all of them may be relevant. In order to select the most relevant paths, we propose a two-step method: (i) we score each vertex with a score (*Vscore*) that reflects its importance in the sub-graph and on the basis of the vertices' *Vscores* we determine a path score *Pscore*, as shown in Figure 3; (ii) we select the top-k paths with respect to the computed path score (*Pscore*).

(i) Vertex Scores and Path Scores: We hypothesize that the most useful commonsense relation paths should include vertices that are *important* with respect to the entire extracted subgraph. We measure the importance of a vertex using different local graph measures: the *closeness centrality measure, page rank* or *personalized page rank*.

Closeness Centrality (CC) (Bavelas, 1950) reflects how close a vertex is to all other vertices in the given graph. It measures the average length of the shortest paths between a given vertex v_i and all other vertices in the given graph G'. In a connected graph, the closeness centrality $CC(v_i)$ of a

vertex $v_i \in G'$ is computed as

$$Vscore_{CC}(v_i) = \frac{|V'|}{\sum_j d(v_j, v_i)} \quad (10)$$

where $|V'|$ represents the number of vertices in the graph G' and $d(v_j, v_i)$ represents the length of the shortest path between v_i and v_j. For each path we compute the normalized sum of $Vscore_X$ of all vertices v_j contained in the path, for any measure $X \in \{CC, PR, PPR\}$.

$$Pscore_X = \frac{\sum_j Vscore_X(v_j)}{N} \quad (11)$$

We rank the paths according to their $Pscore_{CC}$, assuming that relevant paths will contain vertices that are close to the center of the sub-graph G'.

PageRank (PR) (Brin and Page, 1998) is a graph centrality algorithm that measures the relative importance of a vertex in a graph. The Page-Rank score of a vertex $v_i \in G'$ is computed as:

$$Vscore_{PR}(v_i) = \alpha \sum_j u_{ji} \frac{v_j}{L_j} + \frac{1-\alpha}{n} \quad (12)$$

where $L_j = \sum_i u_{ji}$ is the number of neighbors of vertex j, α is a damping factor representing the probability of jumping from a given vertex v_i to another random vertex in the graph and n represents the number of vertices in G'. We calculate $Pscore_{PR}$ using Eq. 11 and order the paths according to their $Pscore_{PR}$, assuming that relevant paths will contain vertices with high relevance, as reflected by a high number of incoming edges.

Personalized PageRank (PPR) (Haveliwala, 2002) is used to determine the importance of a vertex with respect to a certain topic (set of vertices). Instead of assigning equal probability for a random jump $\frac{1-\alpha}{n}$, PPR assigns stronger probability to certain vertices to prefer topical vertices. The PPR score of a vertex $v \in G'$ is computed as:

$$Vscore_{PPR}(v_i) = \alpha \sum_j u_{ji} \frac{v_j}{L_j} + (1-\alpha)T \quad (13)$$

where $T = \frac{1}{|T_j|}$ if nodes v_i belongs to topic T_j and otherwise $T = 0$. In our setting, T_j will contain concepts from the text and human needs, to assign them higher probabilities. We calculate $Pscore_{PPR}$ using Eq. 11 and order the paths according to their scores, assuming that relevant paths should contain vertices holding importance with respect to vertices representing concepts from the text and human needs.
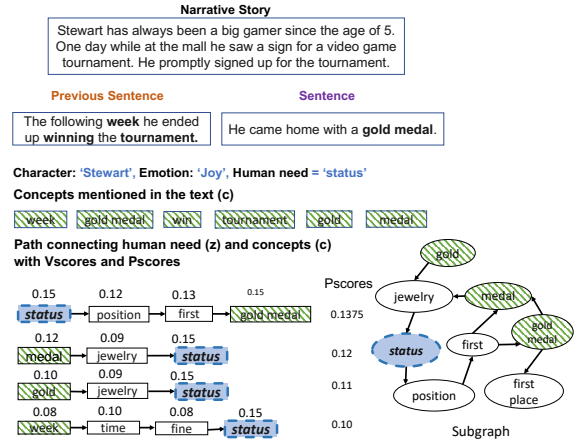

Figure 3: Illustration of commonsense path selection. Top: Context and sentence, Bottom: Selected knowledge paths with *Vscores* and *Pscores* (left) and the corresponding subgraph. Concepts from the text are marked with green dashed lines; blue boxes show the human need label *status* assigned to *Stewart*.

(ii) Path Selection: We rank knowledge paths based on their *Pscore* using the above relevance measures, and construct ranked lists of paths of two types: (i) paths connecting a human needs concept $z \in Z$ to a concept mentioned in the text (p_{c-z}) [2] and (ii) paths connecting concepts in the text (p_{c-c}) [3]. Ranked lists of paths are constructed individually for concepts that constitute the start or endpoint of a path: a human needs concept for p_{c-z} or any concept from the text for p_{c-c}.

Figure 3 illustrates an example where the character *Stewart* felt *joy* after winning a gold medal. The annotated human need label is *status*. We show the paths selected by our algorithm that connect concepts from the text and the human need *status*. We select the top-k paths of type p_{c-z} for each human need to capture relevant knowledge about human needs in relation to concepts in the text. Similarly, we select the top-k paths of type p_{c-c} for each c_i to capture relevant knowledge about the text (not shown in Fig. 3).

3.3 Extending the Model with Knowledge

We have seen how to obtain a ranked list of commonsense knowledge paths from a subgraph extracted from ConceptNet that connect concepts from the textual input and possible human needs categories that are the system's classification tar-

---

[2] p_{c-z} denotes path connecting a human needs concept $z \in Z$ and a concept c mentioned in the text.

[3] p_{c-c} denotes path connecting a concept c and another concept c mentioned in the text.

gets. Our intuition is that the extracted commonsense knowledge paths will provide useful evidence for our model to *link the content expressed in the text to appropriate human need categories*. Paths that are selected by the model as a relevant connection between the input text and the labeled human needs concept can thus provide *explanations* for emotions or goals expressed in the text *in view of a human needs category*. We thus integrate these knowledge paths into our model, (i) to help the model making correct predictions and (ii) to provide explanations of emotions expressed in the text in view of different human needs categories. For each input, we represent the extracted ranked list of n commonsense knowledge paths p as a list $cr^{k,1}, cr^{k,2}, ..., cr^{k,n}$, where each $cr^{k,i}_{1:l}$ represents a path consisting of concepts and relations, with l the length of the path. We embed all concepts and relations in $cr^{k,i}_{1:l}$ with pretrained GloVe (Pennington et al., 2014) embeddings.

Encoding Layer: We use a single-layer BiLSTM to obtain encodings ($h^{k,i}$) for each knowledge path

$$h^{k,i} = BiLSTM(e^{k,i}_{1:n}) \qquad (14)$$

where h^k represents the output of the BiLSTM for the knowledge path and i its the ranking index.

Attention Layer: We use an attention layer, where each encoded commonsense knowledge path interacts with the sentence representation x^s to receive attention weights ($\hat{h}^{k,i}$):

$$\widetilde{h}^{k,i} = \sigma(x^s h^{k,i}), \quad \hat{h}^{k,i} = \frac{\widetilde{h}^{k,i}}{\sum_{i=1}^{N} \widetilde{h}^{k,i}} \qquad (15)$$

In Eq. 15, we use sigmoid to calculate the attention weights, similar to Eq. 6. However, this time we compute attention to highlight which knowledge paths are important for a given input representation (x^s being the final state hidden representation over the input sentence, Eq. 7). To obtain the sentence-aware commonsense knowledge representation x^k, we pass the output of the attention layer through a feedforward layer. W_k, b_k are trainable parameters.

$$x^k = ReLU(W_k(\sum_{i=1}^{N} \hat{h}^{k,i} h^{k,i}) + b_k) \qquad (16)$$

3.4 Distilling Knowledge into the Model

In order to incorporate the selected and weighted knowledge into the model, we concatenate the sentence x^s, context x^{cxt} and knowledge x^k representation and pass it through a FF layer.

$$o_i = ReLU(W_z[x^s_i; x^{cxt}_i; x^k_i] + b_z) \qquad (17)$$

We employ a gating mechanism to allow the model to selectively incorporate relevant information from commonsense knowledge x^k and from the joint input representation y_i (see Eq. 9) *separately*. We finally pass it to a logistic regression classifier to predict a binary label for each class z in the set Z of category labels

$$z_i = \sigma(W_{\widetilde{y}_z}(o_i \odot y_i + o_i \odot x^k_i) + b_{\widetilde{y}_z}) \qquad (18)$$

where $\odot$ represents element-wise multiplication, $b_{\widetilde{y}_z}$, $W_{\widetilde{y}_z}$ are trainable parameters.

4 Experimental Setup

Dataset: We evaluate our model on the *Modeling Naive Psychology of Characters in Simple Commonsense Stories (MNPCSCS)* dataset (Rashkin et al., 2018). It contains narrative stories where each sentence is annotated with a character and a set of human need categories from two inventories: Maslow's (with five coarse-grained) and Reiss's (with 19 fine-grained) categories (Reiss's labels are considered as subcategories of Maslow's). The data contains the original worker annotations. Following prior work we select the annotations that display the "majority label" i.e., categories voted on by ≥ 2 workers. Since no training data is available, similar to prior work we use a portion of the devset as training data, by performing a random split, using 80% of the data to train the classifier, and 20% to tune parameters. Data statistics is reported in Table 1.

Rashkin et al. (2018) report that there is low annotator agreement i.a. between the *belonging* and the *approval* class. We also find high co-occurrence of the *belonging, approval* and *social contact* classes, where *belonging* and *social contact* both pertain to the Maslow class *Love/belonging* while *approval* belongs to the

Classification	Train	Dev	Test
Reiss	5432	1469	5368
Reiss without *belonging* class	5431	1469	5366
Maslow	6873	1882	6821

Table 1: Dataset Statistics: nb. of instances (sentences with annotated characters and human need labels).

Maslow class *Esteem*. This indicates that *belonging* interacts with *Love/belonging* and *Esteem* in relation to social contact. We further observed during our study that in the Reiss dataset the number of instances annotated with the *belonging* class is very low (no. of instances in training is 24, and in dev 5). The performance for this class is thus severely hampered, with 4.7 F_1 score for BiLSTM+Self-Attention and 7.1 F_1 score for BiLSTM+Self-Attention+Knowledge. After establishing benchmark results with prior work (cf. Table 2, including *belonging*), we perform all further experiments with a reduced Reiss dataset, by eliminating the *belonging* class from all instances. This impacts the overall number of instances only slightly: by *one* instance for training and *two* instances for test, as shown in Table 1.

Training: During training we minimize the weighted binary cross entropy loss,

$$L = \sum_{z=1}^{Z} w_z y_z log\widetilde{y}_z + (1 - w_z)(1 - y_z)log(1 - \widetilde{y}_z) \quad (19)$$

$$w_z = \frac{1}{1 - exp^{-\sqrt{P(y_z)}}} \quad (20)$$

where Z is the number of class labels in the classification tasks and w_z is the weight. $P(y_z)$ is the marginal class probability of a positive label for z in the training set.

Embeddings: To compare our model with prior work we experiment with pretrained GloVe (100d) embeddings (Pennington et al., 2014). Otherwise we used GloVe (300d) and pretrained ELMo embeddings (Peters et al., 2018) to train our model.

Hyperparameters for Knowledge Inclusion: We compute ranked lists of knowledge paths of two types: p_{c-z} and p_{c-c}. We use the top-3 p_{c-z} paths for each z using our best ranking strategy (Closeness Centrality + Personalized PageRank) in our best system results (Tables 2, 3, 5), and also considered paths p_{c-c} (top-3 per pair) when evaluating different path selection strategies (Table 4).

Evaluation Metrics: We predict a binary label for each class using a binary classifier so the prediction of each label is conditionally independent of the other classes given a context representation of the sentence. In all prediction tasks we report the micro-averaged Precision (P), Recall (R) and F_1 scores by counting the number of positive instances across all of the categories. All reported results are averaged over five runs. More informa-

Model	WE	Reiss			Maslow		
		P	R	F1	P	R	F1
BiLSTM°	G_{100d}	18.35	27.61	22.05	31.29	33.85	32.52
CNN°	G_{100d}	18.89	31.22	23.54	27.47	41.01	32.09
REN°	G_{100d}	16.79	22.20	19.12	26.24	42.14	32.34
NPN°	G_{100d}	13.13	26.44	17.55	24.27	44.16	31.33
BM	G_{100d}	25.08	28.25	26.57	47.65	60.98	53.54
BM + K♣	G_{100d}	28.47	39.13	32.96	50.54	64.54	56.69
BM	ELMo	29.50	44.28	$35.41_{\pm0.23}$	53.86	67.23	$59.81_{\pm0.23}$
BM + K♣	ELMo	31.74	43.51	$36.70_{\pm0.14}$	57.90	66.07	$61.72_{\pm0.11}$
BM*	ELMo	31.45	44.29	37.70			
BM + K*♣	ELMo	36.76	42.53	39.44			

Table 2: Multi-label Classification Results: °: results in Rashkin et al.; *: w/o *belonging*; BM: BiLSTM+Self-Att.; +K:w/ knowledge, ♣:ranking method CC+PPR.

tion on the dataset, metrics and all other training details are given in the Supplement.

5 Results

Our experiment results are summarized in Table 2. We benchmark our baseline BiLSTM+Self-Attention model (BM, BM w/ knowledge) against the models proposed in Rashkin et al. (2018): a BiLSTM and a CNN model, and models based on the recurrent entity network (REN) (Henaff et al., 2016) and neural process networks (NPN) (Bosselut et al., 2017). The latter differ from the basic encoding models (BiLSTM, CNN) and our own models by explicitly modeling entities. We find that our baseline model BM outperforms all prior work, achieving new state-of-the-art results. For Maslow we show improvement of 21.02 pp. F_1 score. For BM+K this yields a boost of 6.39 and 3.15 pp. F_1 score for Reiss and Maslow, respectively. When using ELMo with BM we see an improvement in recall. However, adding knowledge on top improves the precision by 2.24 and 4.04 pp. for Reiss and Maslow. In all cases, injecting knowledge improves the model's precision and F_1 score.

Table 2 (bottom) presents results for the reduced dataset, after eliminating Reiss' label *belonging*. Since *belonging* is a rare class, we observe further improvements. We see the same trend: adding knowledge improves the precision of the model.

5.1 Model Ablations

To obtain better insight into the contributions of individual components of our models, we perform an ablation study (Table 3). Here and in all later experiments we use richer (300d) GloVe embeddings and the dataset w/o *belonging*. We show results including and not including self-attention

WE	Atten	K	Gated	P	R	F1
G_{300d}	-	-	-	23.31	34.69	27.89
G_{300d}	✓	-	-	26.09	35.59	30.11
G_{300d}	✓	✓	-	27.99	37.73	32.14
G_{300d}	✓	✓	✓	**28.65**	**39.42**	**33.19**
ELMo	-	-	-	32.35	42.66	36.80
ELMo	✓	-	-	31.45	44.29	37.70
ELMo	✓	✓	-	32.65	**45.60**	38.05
ELMo	✓	✓	✓	**36.76**	42.53	**39.44**

Table 3: Model ablations for Reiss Classification on *MNPCSCS* dataset w/o *belonging*.

Path	Ranking	P	R	F1
S+M(P_{c-z}+ P_{c-c})	None	32.51	42.70	36.90
S+M(P_{c-z}+ P_{c-c})	Random	31.63	43.35	36.57
Single Hop(P_{c-z})	CC + PPR	33.00	44.63	37.94
S+M(P_{c-c} + P_{c-z})	CC + PPR	35.30	44.11	39.21
S+M(P_{c-z})	CC	33.45	**47.93**	39.40
S+M(P_{c-z})	PR	35.51	42.82	38.82
S+M(P_{c-z})	PPR	36.23	43.09	39.34
S+M(P_{c-z})	CC + PPR	**36.76**	42.53	**39.44**

Table 4: Results for different path selection strategies on *MNPCSCS* w/o *belonging*; S+M:Single+Multi hop.

and knowledge components. We find that using self-attention over sentences and contexts is highly effective, which indicates that learning how much each token contributes helps the model to improve performance. We observe that integrating knowledge improves the overall F_1 score and yields a gain in precision with ELMo. Further, integrating knowledge using the gating mechanism we see a considerable increase of 3.58 and 1.74 pp. F_1 score improvement over our baseline model for GloVe and ELMo representations respectively.

5.2 Commonsense Path Selection

We further examine model performance for (i) different variants of selecting commonsense knowledge, including (ii) the effectiveness of the relevance ranking strategies discussed in §3.2.2. In Table 4, rows 3-4 use our best ranking method: CC+PPR; rows 5-8 show results when using the top-3 ranked p_{c-z} paths for each human need z with different ranking measures. *None* shows results when no selection is applied to the set of extracted knowledge paths (i.e., using all possible paths from p_{c-z} and p_{c-c}). *Random* randomly selects 3 paths for each human need from the set of paths used in *None*. This yields only a slight drop in performance. This suggests that not every path is relevant. We evaluate the performance when only considering single-hop paths (now top-3 ranked using CC+PPR) *(Single-Hop)*. We see an improvement over random paths and no selection,

but not important enough. In contrast, using both single and multi-hop paths in conjunction with relevance ranking improves the performance considerably (rows 4-8). This demonstrates that multi-hop paths are informative. We also experimented with p_{c-c}+p_{c-z}. We find improvement in recall, however the overall performance decreases by 0.2 F_1 score compared to paths p_{c-z} ranked using CC + PPR. Among different ranking measures *precision* for Personalized PageRank performs best in comparison with CC and PR in isolation, and recall for CC in isolation is highest. Combining CC and PPR yields the best results among the different ranking strategies (rows 5-8).

6 Analysis

6.1 Performance per Human Need Categories

We examined the model performance on each category (cf. Figure 4). The model performs well for basic needs like *food, safety, health, romance*, etc. We note that inclusion of knowledge improves the performance for most classes (only 5 classes do not profit from knowledge compared to only using ELMo), especially for labels which are rare like *honor, idealism, power*. We also found that the annotated labels can be subjective. For instance, *Tom lost his job* is annotated with *order* while our model predicts *savings*, which we consider to be correct. Similar to Rashkin et al. (2018) we observe that preceding context helps the model to better predict the characters' needs, e.g., *Context: Erica's [..] class had a reading challenge [..]. If she was able to read 50 books [..] she won a pizza party!; Sentence: She read a book every day for the entire semester* is annotated with *competition*. Without context the predicted label is *curiosity*, however when including context, the model predicts *competition, curiosity*. We measure the models performance when applying it only to the first sentence of each story (i.e., without the context). As shown in Table 5, also in this setting the inclusion of knowledge improves the performance.

Model	WE	P	R	F1
BM	ELMo	33.39	45.15	38.39
BM+K	ELMo	36.36	44.02	39.83

Table 5: Multi-label classification on *MNPCSCS* w/o *belonging class* and w/o context (1st sentence only)
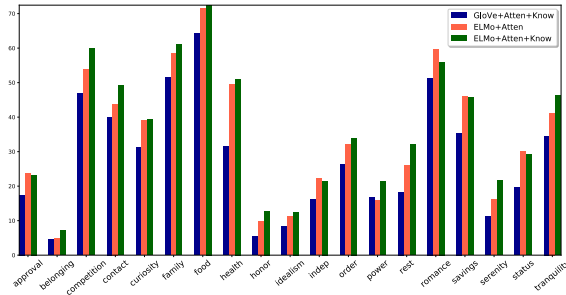
3678

Figure 4: Best model's performance per human needs (F_1 scores) for Reiss on *MNPCSCS* dataset.

Context: Timmy had to renew his driver's license. He went to his local DMV. He waited in line for nearly 2 hours. He took a new picture for his driver's license.
Sentence: He drove back home after an exhausting day.
True Label: *rest*
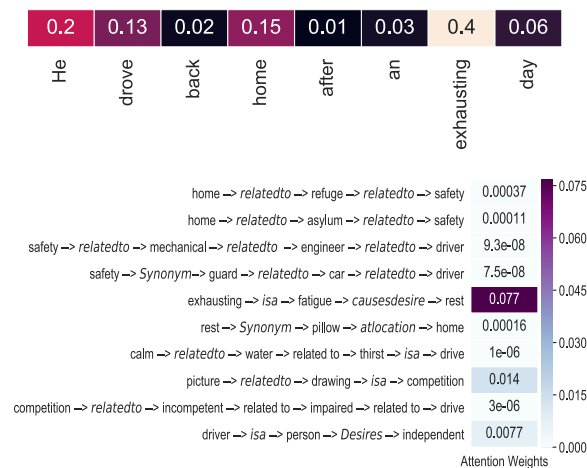**Predicted Label (BM):** *status, approval, order*
Predicted Label (BM+K): *rest*

Figure 5: Interpreting the attention weights on sentence representation and selected commonsense paths.

6.2 Human Evaluation of Extracted Paths

We conduct human evaluation to test the effectiveness and relevance of the extracted commonsense knowledge paths. We randomly selected 50 sentence-context pairs with their gold labels from the devset and extracted knowledge paths that contain the gold label (using CC+PPR for ranking). We asked three expert evaluators to decide whether the paths are relevant to provide information about the missing links between the concepts in the sentence and the human need (gold label). The inter-annotator agreement had a Fleiss' $\kappa = 0.76$. The result for this evaluation shows that in 34% of the cases computed on the basis of majority agreement, our algorithm was able to select a relevant commonsense path. More details about the human evaluation are given in the Supplement.

6.3 Interpretabilty

Finally we study the learned attention distributions of the interactions between sentence representation and knowledge paths, in order to interpret how knowledge is employed to make predictions. Visualization of the attention maps gives evidence of the ability of the model to capture relevant knowledge that connects human needs to the input text. The model provides interpretability in two ways: by selecting tokens from the input text using Eq.6 and by choosing knowledge paths from the imported knowledge using Eq.15 as shown in Figure 5. Figure 5 shows an example where including knowledge paths helped the model to predict the correct human need category. The attention map depicts which exact paths are selected to make the prediction. In this example, the model correctly picks up the token *"exhausting"* from the input sentence and the knowledge path *"exhausting is a fatigue causes desire rest"*. We present more examples of extracted knowledge and its attention visualization in the Supplement.

7 Conclusion

We have introduced an effective new method to rank multi-hop relation paths from a common-sense knowledge resource using graph-based algorithms. Our end-to-end model incorporates multi-hop knowledge paths to predict human needs. Due to the attention mechanism we can analyze the knowledge paths that the model considers in prediction. This enhances transparency and interpretability of the model. We provide quantitative and qualitative evidence of the effectiveness of the extracted knowledge paths. We believe our relevance ranking strategy to select multi-hop knowledge paths can be beneficial for other NLU tasks. In future work, we will investigate structured and unstructured knowledge sources to find explanations for sentiments and emotions.

References

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for Generative Multi-Hop Question Answering Tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230.

Alex Bavelas. 1950. Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America*, 22(6):725–730.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question Answering with Subgraph Embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2017. Simulating action dynamics with neural process networks. *CoRR*, abs/1711.05313.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Eunsol Choi, Hannah Rashkin, Luke Zettlemoyer, and Yejin Choi. 2016. Document-level sentiment inference with social, faction, and discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 333–343.

Lingjia Deng and Janyce Wiebe. 2015. Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 179–189.

Haibo Ding and Ellen Riloff. 2018. Human needs categorization of affective events using labeled and unlabeled data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1919–1929.

Andrew B Goldberg, Nathanael Fillmore, David Andrzejewski, Zhiting Xu, Bryan Gibson, and Xiaojin Zhu. 2009. May all your wishes come true: A study of wishes and how to recognize them. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 263–271.

William L Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 595. NIH Public Access.

Taher H Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2016. Tracking the world state with recurrent entity networks. *CoRR*, abs/1612.03969.

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

Diederik P Kingma and Jimmy Lei Ba. 2014. Adam: Amethod for stochastic optimization. In *Proc. 3rd Int. Conf. Learn. Representations*.

Jiwei Li and Eduard Hovy. 2017. Reflections on sentiment/opinion analysis. In *A Practical Guide to Sentiment Analysis*, pages 41–59. Springer.

Zheng Li, Yu Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2017)*.

Abraham Harold Maslow. 1943. A theory of human motivation. *Psychological review*, 50(4):370.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86.

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

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

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. *Theories of emotion*, 1(4):3–31.

Elahe Rahimtoroghi, Jiaqi Wu, Ruimin Wang, Pranav Anand, and Marilyn Walker. 2017. Modelling protagonist goals and desires in first-person narrative. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 360–369.

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018. Modeling naive psychology of characters in simple commonsense stories. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2289–2299.

Marek Rei and Anders Søgaard. 2018. Zero-shot sequence labeling: Transferring knowledge from sentences to tokens. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 293–302.

Steven Reiss. 2002. *Who am I?: 16 basic desires that motivate our actions define our persona*. Penguin.

Steven Reiss. 2004. Multifaceted nature of intrinsic motivation: The theory of 16 basic desires. *Review of general psychology*, 8(3):179.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Robert Speer and Catherine Havasi. 2012. Representing General Relational Knowledge in ConceptNet 5. In *LREC*, pages 3679–3686.

Srikanth Tamilselvam, Seema Nagar, Abhijit Mishra, and Kuntal Dey. 2017. Graph Based Sentiment Aggregation using ConceptNet Ontology. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 525–535.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Dirk Weissenborn, Tomáš Kočiskỳ, and Chris Dyer. 2017. Dynamic Integration of Background Knowledge in Neural NLU Systems. *arXiv preprint arXiv:1706.02596*.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2017. Incorporating loose-structured knowledge into conversation modeling via recall-gate LSTM. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 3506–3513. IEEE.

Yichun Yin, Yangqiu Song, and Ming Zhang. 2017. Document-level multi-aspect sentiment classification as machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2044–2054.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.