Replicated Siamese LSTM in Ticketing System for Similarity Learning and Retrieval in Asymmetric Texts

Pankaj Gupta[1,2] **Bernt Andrassy**[1] **Hinrich Schütze**[2]
[1]Corporate Technology, Machine-Intelligence (MIC-DE), Siemens AG Munich, Germany
[2]CIS, University of Munich (LMU) Munich, Germany
pankaj.gupta@siemens.com | bernt.andrassy@siemens.com
pankaj.gupta@campus.lmu.de | inquiries@cis.lmu.de

Abstract

The goal of our industrial ticketing system is to retrieve a relevant solution for an input query, by matching with historical tickets stored in knowledge base. A query is comprised of subject and description, while a historical ticket consists of subject, description and solution. To retrieve a relevant solution, we use textual similarity paradigm to learn similarity in the query and historical tickets. The task is challenging due to significant term mismatch in the query and ticket pairs of asymmetric lengths, where subject is a short text but description and solution are multi-sentence texts. We present a novel Replicated Siamese LSTM model to learn similarity in asymmetric text pairs, that gives 22% and 7% gain (Accuracy@10) for retrieval task, respectively over unsupervised and supervised baselines. We also show that the topic and distributed semantic features for short and long texts improved both similarity learning and retrieval.

1 Introduction

Semantic Textual Similarity (STS) is the task to find out if the text pairs mean the same thing. The important tasks in Natural Language Processing (NLP), such as Information Retrieval (IR) and text understanding may be improved by modeling the underlying semantic similarity between texts.

With recent progress in deep learning, the STS task has gained success using LSTM (Mueller and Thyagarajan, 2016) and CNN (Yin et al., 2016) based architectures; however, these approaches model the underlying semantic similarity between example pairs, each with a single sentence or phrase with term overlaps. In the domain of question retrieval (Cai et al., 2011; Zhang et al., 2014), users retrieve historical questions which precisely match their questions (single sentence) semantically equivalent or relevant. However, we investigate similarity learning between texts of asymmetric lengths, such as short (phrase) Vs longer (paragraph/documents) with significant term mismatch. The application of textual understanding in retrieval becomes more challenging when the relevant document-sized retrievals are stylistically distinct with the input short texts. Learning a similarity metric has gained much research interest, however due to limited availability of labeled data and complex structures in variable length sentences, the STS task becomes a hard problem. The performance of IR system is sub-optimal due to significant term mismatch in similar texts (Zhao, 2012), limited annotated data and complex structures in variable length sentences. We address the challenges in a real-world industrial application.

Our ticketing system (Figure 1(a)) consists of a query and historical tickets (Table 1). A query (reporting issue, q) has 2 components: *subject* (SUB) and *description* (DESC), while a historical ticket ($\mathfrak{t}$) stored in the knowledge base (KB) has 3 components: SUB, DESC and *solution* (SOL). A SUB is a short text, but DESC and SOL consist of multiple sentences. Table 1 shows that SUB $\in q$ and SUB $\in \mathfrak{t}$ are semantically similar and few terms in SUB $\in q$ overlap with DESC $\in \mathfrak{t}$. However, the expected SOL $\in \mathfrak{t}$ is distinct from both SUB and DESC $\in q$. The goal is to retrieve an optimal action (i.e. SOL from $\mathfrak{t}$) for the input q.

To improve retrieval for an input q, we adapt the Siamese LSTM (Mueller and Thyagarajan, 2016) for similarity learning in asymmetric text pairs, using the available information in q and $\mathfrak{t}$. For instance,

Proceedings of SemDeep-3, the 3rd Workshop on Semantic Deep Learning, pages 1–11
Santa Fe, New Mexico, USA, August 20, 2018.

QUERY (q)

SUB: GT Trip - Low Frequency Pulsations

DESC: GT Tripped due to a sudden increase in Low Frequency Pulsations. The machine has been restarted and is now operating normally. Alarm received was: GT XXX Low Frequency Pulsation.

HISTORICAL TICKET (t)

SUB: Narrow Frequency Pulsations

DESC: Low and Narrow frequency pulsations were detected. The peak value for the Low Frequency Pulsations is ## mbar.

SOL: XXXX combustion support is currently working on the issue. The action is that the machine should not run until resolved.

Table 1: Example of a Query and Historical Ticket

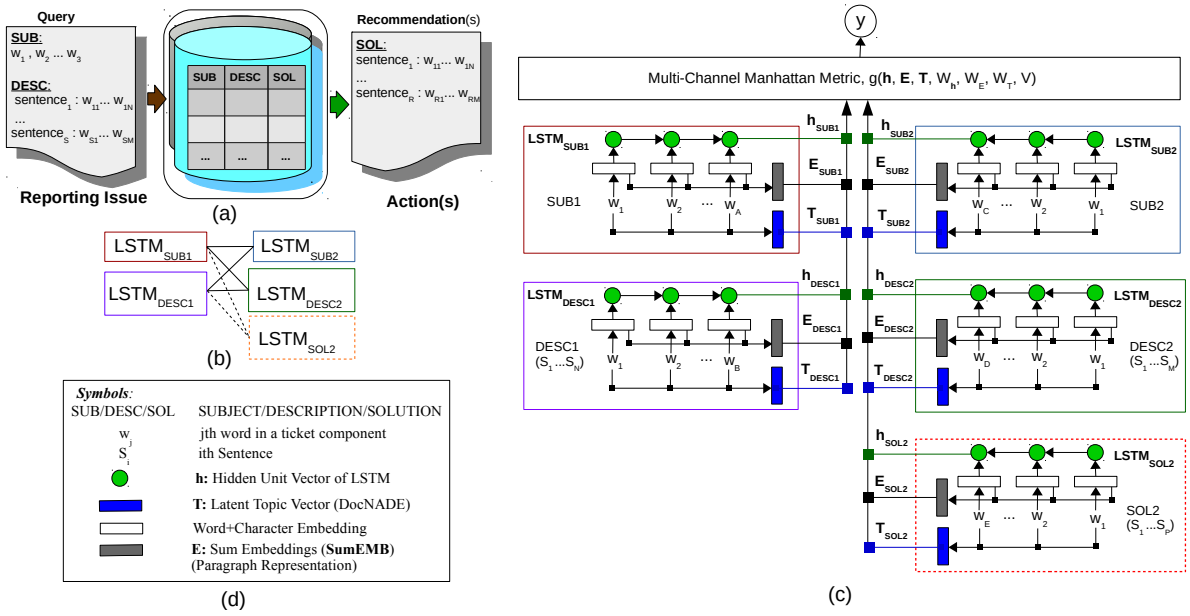

Figure 1: (a): Intelligent Ticketing System (ITS) (b): High-level illustration of Siamese LSTM for cross-level pairwise similarity. (c): Replicated Siamese with multi-channel (SumEMB, LSTM and topic vectors) and multi-level (SUB, DESC and/or SOL) inputs in the objective function, g. y: similarity score. The dotted lines indicate ITS output. (d): Symbols used.

we compute *multi-level* similarity between (SUB $\in q$, SUB $\in$ t) and (DESC $\in q$, DESC $\in$ t). However, observe in Table 1 that the *cross-level* similarities such as between (SUB $\in q$, DESC $\in$ t), (DESC $\in q$, SUB $\in$ t) or (SUB $\in q$, SOL $\in$ t), etc. can supplement IR performance. See Figure 1(b).

The *contributions* of this paper are as follows: (1) Propose a novel architecture (Replicated Siamese LSTM) for similarity learning in asymmetric texts via multi-and-cross-level semantics (2) Investigate distributed and neural topic semantics for similarity learning via multiple channels (3) Demonstrate a gain of 22% and 7% in Accuracy@10 for retrieval, respectively over unsupervised and supervised baselines in the industrial application of a ticketing system.

2 Methodology

Siamese networks (Chopra et al., 2005) are dual-branch networks with tied weights and an objective function. The aim of training is to learn text pair representations to form a highly structured space where they reflect complex semantic relationships. Figure 1 shows the proposed Replicated Siamese neural network architecture such that ($\text{LSTM}_{\text{SUB1}}$+$\text{LSTM}_{\text{DESC1}}$) = ($\text{LSTM}_{\text{SUB2}}$+$\text{LSTM}_{\text{DESC2}}$+$\text{LSTM}_{\text{SOL2}}$), to learn similarities in asymmetric texts, where a query (SUB1+DESC1) is stylistically distinct from a historical ticket (SUB2+DESC2+SOL2).

Note, the *query components are suffixed by "1" and historical ticket components by "2"* in context of the following work for pairwise comparisons.

$$g(h, E, T, W_h, W_E, W_T, V) =$$
$$\exp\Big(-\sum_{p \in \{SUB1, DESC1\}} \sum_{q \in \{SUB2, DESC2, SOL2\}} V_{\{p,q\}} \left(W_h \|h_p - h_q\|_1 + W_E \|E_p - E_q\|_1 + W_T \|T_p - T_q\|_1\right)\Big) \quad (1)$$

Figure 2: Multi-Channel Manhattan Metric

2.1 Replicated, Multi-and-Cross-Level, Multi-Channel Siamese LSTM

Manhattan LSTM (Mueller and Thyagarajan, 2016) learns similarity in text pairs, each with a single sentence; however, we advance the similarity learning task in asymmetric texts pairs consisting of one or more sentences, where similarity is computed between different-sized subject and description or solution texts. As the backbone of our work, we compute similarity scores to learn a highly structured space via LSTM (Hochreiter and Schmidhuber, 1997) for representation of each pair of the query (SUB1 and DESC1) or historical ticket (SUB2, DESC2 and SOL2) components, which includes multi-level (SUB1-SUB2, DESC1-DESC2) and cross-level (SUB1-DESC2, SUB1-SOL2, etc.) asymmetric textual similarities, Figure 1(b) and (c). To accumulate the semantics of variable-length sentences $(x_1, ..., x_T)$, recurrent neural networks (RNNs) (Vu et al., 2016a; Gupta et al., 2016; Gupta and Andrassy, 2018), especially the LSTMs (Hochreiter and Schmidhuber, 1997) have been successful.

LSTMs are superior in learning long range dependencies through their memory cells. Like the standard RNN (Mikolov et al., 2010; Gupta et al., 2015a; Vu et al., 2016b), LSTM sequentially updates a hidden-state representation, but it introduces a memory state c_t and three gates that control the flow of information through the time steps. An output gate o_t determines how much of c_t should be exposed to the next node. An input gate i_t controls how much the input x_t be stored in memory, while the forget gate f_t determines what should be forgotten from memory. The dynamics:

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1}) \\ f_t &= \sigma(W_f x_t + U_f h_{t-1}) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1}) \\ \tilde{c}_t &= tanh(W_c x_t + U_c h_{t-1}) \\ c_t &= i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \quad (2)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ and $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. The proposed architecture, Figure 1(c) is composed of multiple uni-directional LSTMs each for subject, description and solution within the Siamese framework, where the weights at over levels are shared between the left and right branch of the network. Therefore, the name *replicated*.

Each LSTM learns a mapping from space of variable length sequences, including asymmetric texts, to a hidden-state vector, h. Each sentence $(w_1, ...w_T)$ is passed to LSTM, which updates hidden state via eq 2. A final encoded representation (e.g. h_{SUB1}, h_{SUB2} in Figure 1(c)) is obtained for each query or ticket component. A single LSTM is run over DESC and SOL components, consisting of one or more sentences. Therefore, the name *multi-level* Siamese.

The representations across the text components (SUB DESC or SOL) are learned in order to maximize the similarity and retrieval for a query with the historical tickets. Therefore, the name *cross-level* Siamese.

The sum-average strategy over word embedding (Mikolov et al., 2010) for short and longer texts has demonstrated a strong baseline for text classification (Joulin et al., 2016) and pairwise similarity learning (Wieting et al., 2016). This simple baseline to represent sentences as bag of words (BoW) inspires us to use the BoW for each query or historical ticket component, for instance E_{SUB1}. We refer the approach as *SumEMB* in the context of this paper.

We supplement the similarity metric (g) with *SumEMB* (E), latent topic (T) (section 2.2) and hidden vectors (h) of LSTM for each text component from both the Siamese branches. Therefore, the name *multi-channel* Siamese.

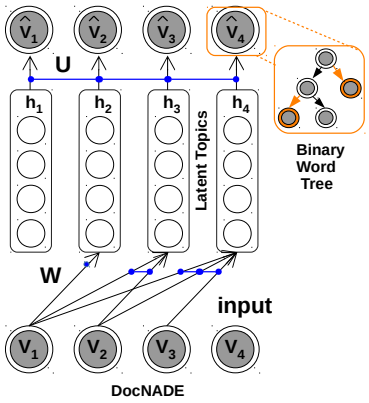

Figure 3: DocNADE: Neural Auto-regressive Topic Model

Parameter	*Search*	*Optimal*
E	[350]	350
T	[20, 50, 100]	100
h	[50, 100]	50
W_h	[0.6, 0.7, 0.8]	0.7
W_E	[0.3, 0.2, 0.1]	0.1
W_T	[0.3, 0.2, 0.1]	0.2
$V_{SUB1-SUB2}$	[0.3, 0.4	0.3
$V_{DESC1-DESC2}$	[0.3, 0.4]	0.3
$V_{SUB1-DESC2}$	[0.10, 0.15, 0.20]	0.20
$V_{SUB1-SOL2}$	[0.10, 0.15, 0.20]	0.10
$V_{DESC1-SOL2}$	[0.10, 0.15, 0.20]	0.10

Table 2: Hyperparameters in the Replicated Siamese LSTM (experiment #No:22)

2.2 Neural Auto-Regressive Topic Model

Topic models such as Latent Dirichlet allocation (LDA) (Blei et al., 2003) and Replicated Softmax (RSM) (Hinton and Salakhutdinov, 2009; Gupta et al., 2018c) have been popular in learning meaningful representations of unlabeled text documents. Recently, a new type of topic model called the Document Neural Autoregressive Distribution Estimator (DocNADE) (Larochelle and Lauly, 2012; Zheng et al., 2016; Gupta et al., 2018a) was proposed and demonstrated the state-of- the-art performance for text document modeling. DocNADE models are advanced variants of Restricted Boltzmann Machine (Hinton, 2002; Salakhutdinov et al., 2007; Gupta et al., 2015b; Gupta et al., 2015c), and have shown to outperform LDA and RSM in terms of both log-likelihood of the data and document retrieval. In addition, the training complexity of a DocNADE model scales logarithmically with vocabulary size, instead linear as in RSM. The features are important for an industrial task along with quality performance. Therefore, we adopt DocNADE model for learning latent representations of tickets and retrieval in unsupervised fashion. See Larochelle and Lauly (2012) and Gupta et al. (2018a) for further details, and Figure 3 for the DocNADE architecture, where we extract the last hidden topic layer (h_4) to compute document representation.

2.3 Multi-Channel Manhattan Metric

Chopra et al. (2005) indicated that using l_2 instead of l_1 norm in similarity metric can lead to undesirable plateaus. Mueller and Thyagarajan (2016) showed stable and improved results using Manhattan distance over cosine similarity.

Mueller and Thyagarajan (2016) used a Manhattan metric (l_1-norm) for similarity learning in single sentence pairs. However, we adapt the similarity metric for 2-tuple (SUB1, DESC1) vs 3-tuple (SUB2, DESC2 and SOL2) pairs, where the error signals are back-propagated in the multiple levels and channels during training to force the Siamese network to entirely capture the semantic differences across the query and historical tickets components. The similarity metric, $g \in [0,1]$ is given in eq 1, where $||\cdot||$ is l_1 norm. W_h, W_E and W_T are the three channels weights for h, E and and T, respectively. The weights (V) are the multi-level weights between the ticket component pairs. Observe that a single weight is being used in the ordered ticket component pairs, for instance $V_{SUB1-DESC2}$ is same as $V_{DESC2-SUB1}$.

3 Evaluation and Analysis

We evaluate the proposed method on our industrial data for textual similarity learning and retrieval tasks in the ticketing system. Table 4 shows the different model configurations used in the following exper-

Held-out	*Perplexity* (100 topics)			
Ticket	$M1$: SUB+DESC		$M2$: SUB+DESC+SOL	
Component	LDA	DocNADE	LDA	DocNADE
DESC	380	**362**	565	**351**
SUB+DESC	480	**308**	515	**289**
SUB+DESC+SOL	553	**404**	541	**322**

(a)

Query Component	Perplexity (100 topics)			
	DocNADE:**M1**		DocNADE:**M2**	
	$\|Q\|_L$	$\|Q\|_U$	$\|Q\|_L$	$\|Q\|_U$
DESC1	192	177	132	118
SUB1+DESC1	164	140	130	118

(b)

Table 3: **(a)** Perplexity by DocNADE and LDA trained with $M1$: SUB+DESC or $M2$: SUB+DESC+SOL on all tickets and evaluated on 50 held-out tickets with their respective components or their combination. Observe that when DocNADE is trained with SUB+DESC+SOL, it performs better when training with SUB+DESC+SOL and outperforms LDA. **(b)** Perplexity by DocNADE: $M1$ trained on SUB+ DESC and $M2$ on SUB+DESC+SOL of the historical tickets.

Model	Model Configuration
T ($X1$–$X2$)	Compute Similarity using topic vector (*T*) pairs of a query ($X1$) and historical ticket ($X2$) components
E ($X1$–$X2$)	Compute Similarity using embedding vector (*E*) pairs of a query ($X1$) and historical ticket ($X2$) components
$X + Y + Z$	Merge text components (SUB, DESC or SOL), representing a single document
T ($X1 + Y1$–$X2 + Y2 + Z2$)	Compute Similarity using topic vector (*T*) pairs of a query ($X1 + Y1$) and historical ticket ($X2 + Y2 + Z2$) components
S-LSTM ($X1$–$X2$)	Compute Similarity using Standard Siamese LSTM on a query ($X1$) and historical ticket ($X2$) components
ML ($X1$–$X2$, $Y1$–$Y2$)	Multi-level Replicated Siamese LSTM. Compute similarity in ($X1$–$X2$) and ($Y1$–$Y2$) components of a query and historical ticket
CL (X, Y, Z)	Cross-level Replicated Siamese LSTM. Compute similarity in ($X1$–$Y2$), ($X1$–$Z2$), ($Y1$–$X2$) and ($Y1$–$Z2$) pairs

Table 4: Different model configurations for the experimental setups and evaluations. See Figure 1(c) for LSTM configurations.

imental setups. We use Pearson correlation, Spearman correlation and Mean Squared Error[1] (MSE) metrics for STS and 9 different metrics (Table 5) for IR task.

3.1 Industrial Dataset for Ticketing System

Our industrial dataset consist of queries and historical tickets. As shown in Table 1, a query consists of *subject* and *description* texts, while a historical ticket in knowledge base (KB) consists of *subject*, *description* and *solution* texts. The goal of the ITS is to automatically recommend an optimal action i.e. *solution* for an input query, retrieved from the existing KB.

There are $\mathfrak{T} = 949$ historical tickets in the KB, out of which 421 pairs are labeled with their relatedness score. We randomly split the labeled pairs by 80-20% for train (P_{tr}) and development (P_{dev}). The relatedness labels are: *YES* (similar that provides correct solution), *REL* (does not provide correct solution, but close to a solution) and *NO* (not related, not relevant and provides no correct solution). We convert the labels into numerical scores [1,5], where *YES*:5.0, *REL*:3.0 and *NO*:1.0. The average length (#words) of SUB, DESC and SOL are 4.6, 65.0 and 74.2, respectively.

The end-user (customer) additionally supplies 28 unique queries (Q_U) (exclusive to the historical tickets) to test system capabilities to retrieve the optimal solution(s) by computing 28×949 pairwise ticket similarities. We use these queries for the end-user qualitative evaluation for the 28×10 proposals (top 10 retrievals for each query).

3.2 Experimental Setup: Unsupervised

We establish baseline for similarity and retrieval by the following two unsupervised approaches:

(1) Topic Semantics T: As discussed in section 2.2, we use DocNADE topic model to learn document representation. To train, we take 50 held-out samples from the historical tickets $\mathfrak{T}$. We compute perplexity on 100 topics for each ticket component from the held-out set, comparing LDA and DocNADE models trained individually with SUB+DESC ($M1$) and SUB+DESC+SOL texts[2] ($M2$). Table 3a shows that DocNADE outperforms LDA.

[1] http://alt.qcri.org/semeval2016/task1/

[2] +: merge texts to treat them as a single document

#No	Model (Query-Historical Ticket)	Similarity Task			Retrieval Task								
		r	ρ	MSE	MAP@1	MAP@5	MAP@10	MRR@1	MRR@5	MRR@10	Acc@1	Acc@5	Acc@10
1	T (SUB1–SUB2) (**unsupervised baseline**)	0.388	0.330	5.122	0.08	0.08	0.07	1.00	0.28	0.10	0.04	0.19	0.30
2	T (SUB1–DESC2)	0.347	0.312	3.882	0.09	0.07	0.07	0.00	0.05	0.08	0.04	0.13	0.21
3	T (DESC1–SUB2)	0.321	0.287	3.763	0.08	0.09	0.09	0.00	0.05	0.11	0.03	0.20	0.31
4	T (DESC1–DESC2)	0.402	0.350	3.596	0.08	0.08	0.08	0.00	0.04	0.10	0.03	0.19	0.33
5	T (SUB1–SUB2+DESC2)	0.413	0.372	3.555	0.09	0.09	0.08	0.00	0.05	0.11	0.04	0.20	0.32
6	T (SUB1+DESC1–SUB2)	0.330	0.267	3.630	0.09	0.10	0.09	0.00	0.26	0.12	0.04	0.23	0.35
7	T (SUB1+DESC1–DESC2)	0.400	0.350	3.560	0.07	0.08	0.08	0.00	0.00	0.10	0.03	0.19	0.35
8	T (SUB1+DESC1–SUB2+DESC2)	0.417	0.378	3.530	0.05	0.07	0.08	0.00	0.07	0.11	0.03	0.22	0.37
9	T (SUB1+DESC1–SUB2+DESC2+SOL2)	0.411	0.387	3.502	0.09	0.09	0.08	0.00	0.06	0.12	0.04	0.20	0.40
11	E (SUB1–SUB2) (**unsupervised baseline**)	0.141	0.108	3.636	0.39	0.38	0.36	0.00	0.03	0.08	0.02	0.13	0.24
12	E (DESC1–DESC2)	0.034	0.059	4.201	**0.40**	**0.40**	**0.39**	0.00	0.10	0.07	0.03	0.12	0.18
13	E (SUB1+DESC1–SUB2+DESC2)	0.103	0.051	5.210	0.16	0.16	0.15	0.00	0.03	0.11	0.07	0.16	0.20
14	E (SUB1+DESC1–SUB2+DESC2+SOL2)	0.063	0.041	5.607	0.20	0.17	0.16	0.00	0.03	0.13	0.05	0.13	0.22
15	S-LSTM(SUB1–SUB2) (**supervised baseline**)	0.530	0.501	3.778	0.272	0.234	0.212	0.000	0.128	0.080	0.022	0.111	0.311
16	S-LSTM (DESC1–DESC2)	0.641	0.586	3.220	0.277	0.244	0.222	0.100	**0.287**	0.209	0.111	0.3111	0.489
17	S-LSTM (SUB1+DESC1–SUB2+DESC2)	0.662	0.621	2.992	0.288	0.251	0.232	0.137	0.129	0.208	0.111	0.342	0.511
18	S-LSTM (SUB1+DESC1–SUB2+DESC2+SOL2)	0.693	0.631	2.908	0.298	0.236	0.241	0.143	0.189	0.228	0.133	0.353	0.548
19	ML-LSTM (SUB1–SUB2, DESC1–DESC2)	0.688	0.644	2.870	0.290	0.255	0.234	0.250	0.121	0.167	0.067	0.289	0.533
20	+ CL-LSTM (SUB, DESC, SOL)	0.744	0.680	2.470	0.293	0.259	0.238	0.143	0.179	0.286	**0.178**	0.378	0.564
21	+ weighted channels (h*0.8, E*0.2)	0.758	0.701	2.354	0.392	0.376	0.346	**0.253**	0.176	0.248	0.111	0.439	0.579
22	+ weighted channels (h*0.7, E*0.1, T*0.2)	**0.792**	**0.762**	**2.052**	0.382	0.356	0.344	0.242	0.202	**0.288**	0.133	**0.493**	**0.618**

Table 5: Results on Development set: Pearson correlation (*r*), Spearmans rank correlation coefficient (ρ), Mean Squared Error (MSE), Mean Average Precision@k (MAP@k), Mean Reciprocal Rank@k (MRR@k) and Accuracy@k (Acc@k) for the multi-level (ML) and cross-level (CL) similarity learning, and retrieving the k-most similar tickets for each query (SUB1+DESC1). **#[1-14]**: Unsupervised baselines with DocNADE (T) and SumEMB (E). **#[15-18]**: Supervised Standard Siamese baselines. **#[19-22]**: Supervised Replicated Siamese with multi-channel and cross-level features.

Model	Similarity Task			Retrieval Task								
	r	ρ	MSE	MAP@1	MAP@5	MAP@10	MRR@1	MRR@5	MRR@10	Acc@1	Acc@5	Acc@10
T (SUB1-SUB2)	0.414	0.363	5.062	0.04	0.03	0.03	**0.29**	0.24	0.10	0.01	0.17	0.28
T (SUB1-DESC2)	0.399	0.362	3.791	0.04	0.03	0.03	0.00	0.05	0.07	0.03	0.12	0.19
T (DESC1-SUB2)	0.371	0.341	3.964	0.05	0.06	0.05	0.25	0.07	0.11	0.04	0.21	0.33
T (DESC1-DESC2)	0.446	0.398	3.514	0.05	0.05	0.04	0.00	0.04	0.10	0.04	0.18	0.34
T (SUB1-SUB2+DESC2)	0.410	0.370	3.633	0.05	0.04	0.04	0.00	0.12	0.08	0.04	0.13	0.20
T (SUB1+DESC2-SUB2)	0.388	0.326	3.561	0.06	0.06	0.05	0.25	**0.29**	0.13	0.05	0.22	0.38
T (SUB1+DESC1-DESC2)	0.443	0.396	3.477	0.04	0.04	0.04	0.00	0.00	0.10	0.03	0.17	0.37
T (SUB1+DESC1, SUB2+DESC2)	**0.466**	**0.417**	3.460	0.05	0.05	0.04	0.00	0.06	0.11	0.03	**0.24**	0.37
T (SUB1+DESC1, SUB2+DESC2+SOL2)	0.418	0.358	**3.411**	**0.07**	0.06	**0.06**	0.00	0.09	**0.14**	0.05	0.20	**0.39**

Table 6: DocNADE ($M2$) performance for the queries $Q_L \in (P_{tr} + P_{dev})$ in the labeled pairs in unsupervised fashion.

Next, we need to determine which DocNADE model ($M1$ or $M2$) is less perplexed to the queries. Therefore, we use $M1$ and $M2$ to evaluate DESC1 and SUB1+DESC1 components of the two sets of queries: (1) Q_L is the set of queries from labeled (421) pairs and (2) Q_U is the end-user set. Table 3b shows that $M2$ performs better than $M1$ for both the sets of queries with DESC1 or SUB1+DESC1 texts. We choose $M2$ version of the DocNADE to setup baseline for the similarity learning and retrieval in unsupervised fashion.

To compute a similarity score for the given query q and historical ticket $\mathfrak{t}$ where $(q, \mathfrak{t}) \in P_{dev}$, we first compute a latent topic vector (*T*) each for q and $\mathfrak{t}$ using DocNADE ($M2$) and then apply the similarity metric g (eq 1). To evaluate retrieval for q, we retrieve the top 10 similar tickets, ranked by the similarity scores on their topic vectors. Table 5 (#No [1-9]) shows the performance of DocNADE for similarity and retrieval tasks. Observe that #9 achieves the best MSE (3.502) and Acc@10 (0.40) out of [1-9], suggesting that the topic vectors of query (SUB1+DESC1) and historical ticket (SUB2+DESC2+SOL2) are the key in recommending a relevant SOL2. See the performance of DocNADE for all labeled pairs i.e. queries and historical tickets ($P_{tr} + P_{dev}$) in the Table 6.

(2) Distributional Semantics E: Beyond topic models, we establish baseline using the SumEMB method (section 2.1), where an embedding vector *E* is computed following the topic semantics approach. The experiments #11-14 show that the SumEMB results in lower performance for both the tasks, suggesting a need of a supervised paradigm in order to learn similarities in asymmetric texts. Also, the comparison with DocNADE indicates that the topic features are important in the retrieval of tickets.

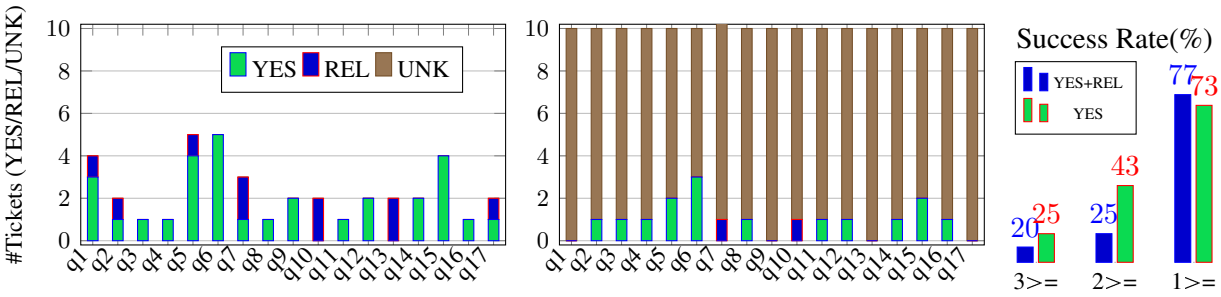

Figure 4: Evaluation on End-user Queries (sub-sample). UNK: Unknown. (Left) Gold Data: The count of similar (YES) and relevant (REL) tickets for each query (q1-q17). (Middle) ITS Results: For each query, ITS proposes the top 10 YES/REL retrievals. The plot depicts the count of YES/REL proposals matched out of the top 10 gold proposals for each q. UNK may include YES, REL or NO, not annotated in the gold pairs. (Right) Success Rate: YES: percentage of correct similar(YES) proposal out of the top 10; YES+REL: percentage of correct similar (YES) and relevant(REL) proposals out of the top 10.

3.3 Experimental Setup: Supervised

For semantic relatedness scoring, we train the Replicated Siamese, using backpropagation-through-time under the Mean Squared Error (MSE) loss function (after rescaling the training-set relatedness labels to lie $\in$ [0, 1]). After training, we apply an additional non-parametric regression step to obtain better-calibrated predictions $\in$ [1, 5], same as (Mueller and Thyagarajan, 2016). We then evaluate the trained model for IR task, where we retrieve the top 10 similar results (SUB2+DESC2+SOL2), ranked by their similarity scores, for each query (SUB1+DESC1) in the development set and compute MAP@K, MRR@K and Acc@K, where K=1, 5, and 10.

We use 300-dimensional pre-trained *word2vec*[3] embeddings for input words, however, to generalize beyond the limited vocabulary in *word2vec* due to industrial domain data with technical vocabulary, we also employ char-BLSTM (Lample et al., 2016) to generate additional embeddings (=50 dimension[4]).The resulting dimension for word embeddings is 350. We use 50-dimensional hidden vector, h_t, memory cells, c_t and Adadelta (Zeiler, 2012) with dropout and gradient clipping (Pascanu et al., 2013) for optimization. The topics vector (T) size is 100. We use python NLTK toolkit[5] for sentence tokenization. See Table 2 for the hyperparameters in Replicated Siamese LSTM for experiment #No:22.

3.4 Results: State-of-the-art Comparisons

Table 5 shows the similarity and retrieval scores for unsupervised and supervised baseline methods. The #9, #18 and #20 show that the supervised approach performs better than unsupervised topic models. #17 and #19 suggest that the multi-level Siamese improves (Acc@10: 0.51 vs. 0.53) both STS and IR. Comparing #18 and #20, the cross-level Siamese shows performance gains (Acc@10: 0.55 vs. 0.57). Finally, #21 and #22 demonstrates improved similarity (MSE: 2.354 vs. 2.052) and retrieval (Acc@10: 0.58 vs. 0.62) due to weighted multi-channel (h, E and T) inputs.

The replicated Siamese (#22) with different features best results in 2.052 for MSE and 0.618 (= 61.8%) for Acc@10. We see 22% and 7% gain in Acc@10 for retrieval task, respectively over unsupervised (#9 vs. #22: 0.40 vs. 0.62) and supervised (#18 vs. #22: 0.55 vs. 0.62) baselines. The experimental results suggest that the similarity learning in supervised fashion improves the ranking of relevant tickets.

3.5 Success Rate: End-User Evaluation

We use the trained similarity model to retrieve the top 10 similar tickets from KB for each end-user query Q_U, and compute the number of correct similar and relevant tickets. For ticket ID $q6$ (Figure 4, Middle),

[3]Publicly available at: code.google.com/p/word2vec
[4]Run forward-backward character LSTM for every word and concatenate the last hidden units (25 dimension each)
[5]http://www.nltk.org/api/nltk.tokenize.html

Query	Recommendation_1	Recommendation_2	Recommendation_3
SUB: GT Trip - Low Frequency Pulsations **DESC:** GT Tripped due to a sudden increase in Low Frequency Pulsations. The machine has been restarted and is now operating normally. Alarm received was: GT XXX Low Frequency Pulsation	**SUB:** Narrow Frequency Pulsations **DESC:** Low and Narrow frequency pulsations were detected. The peak value for the Low Frequency Pulsations is ## mbar. **SOL:** XXXXX combustion support is currently working on the issue. The recommended action for now is that the machine XXXX at **load** XXXX ## **MW**.	**SUB:** Low frequency pulsations **DESC:** High level low frequency pulsations were detected when active **load** is XXXX. **SOL:** Since the machine is running with XXXX, the XXX be changed in the register. After adjustment is complete, monitor the machine behavior between ## **MW** to ## **load**.	**SUB:** GT3 - High Low Frequency Pulsation alarms after trip **DESC:** Yesterday, after Steam Turbine tripped, GT-3 experienced high Low Frequency Pulsation alarm. The**load** of GT-3 was ## **MW** and went up as high as ## **MW**. During the time, Low Frequency Pulsation for 3 pulsation devices went up as high as ##. The Low frequency pulsation was a XXX. **SOL:** A **load** XXXX from ## **MW** to ## **MW** is an event XXX the unit XXXX trip. The XXXX to low frequency pulsation during similar event, should be XXXX. Check that XXXX from after the XXXX (XX005/XX01) into combustion chamber (XX030/XX01), XXXX should be XXXX. Repeat until XXXX is within the range of ## -##.
(Rank, Similarity Score)	(1, 4.75)	(2, 4.71)	(3, 4.60)
#Topics {**#83**, #7, #30}	{**#83**, #16, #30}	{#7, **#83**, #19}	{#7, **#83**, #19}

Table 7: Top-3 Tickets Retrieved and ordered by their (rank, similarity score) for an input test query. $\#Topics$: the top 3 most probable associated topics. **SOL** of the retrieved tickets is returned as recommended action. Underline: Overlapping words; XXXX and ##: Confidential text and numerical terms.

3 out of 10 proposed tickets are marked similar, where the end-user expects 4 similar tickets (Figure 4, Left). For ticket ID $q1$, $q13$ and $q17$, the top 10 results do not include the corresponding expected tickets due to no term matches and we find that the similarity scores for all the top 10 tickets are close to 4.0 or higher, which indicates that the system proposes more similar tickets (than the expected tickets), not included in the gold annotations. The top 10 proposals are evaluated for each query by success rate (success, if N/10 proposals supply the expected solution). We compute success rate (Figure 4, Right) for (1 or more), (2 or more) and (3 or more) correct results out of the top 10 proposals.

4 Qualitative Inspections for STS and IR

Table 7 shows a real example for an input query, where the top 3 recommendations are proposed from the historical tickets using the trained Replicated Siamese model. The recommendations are ranked by their similarity scores with the query. The underline shows the overlapping texts.

We also show the most probable topics (#) that the query or each recommendation is associated with. The topics shown (Table 8) are learned from DocNADE model and are used in multi-channel network. Observe that the improved retrieval scores (Table 5 #22) are attributed to the overlapping topic semantics in query and the top retrievals. For instance, the topic #83 is the most probable topic feature for the query and recommendations. We found terms, especially *load* and *MW* in SOL (frequently appeared for other *Frequency Pulsations* tickets) that are captured in topics #7 and #83, respectively.

5 Related Work

Semantic Textual Similarity has diverse applications in information retrieval (Larochelle and Lauly, 2012; Gupta et al., 2018a), search, summarization (Gupta et al., 2011), recommendation systems, etc. For shared STS task in SemEval 2014, numerous researchers applied competitive methods that utilized both heterogeneous features (e.g. word overlap/similarity, negation modeling, sentence/phrase composition) as well as external resources (e.g. Wordnet (Miller, 1995)), along with machine learning approaches such as LSA (Zhao et al., 2014) and word2vec neural language model (Mikolov et al., 2013). In the domain of question retrieval (Cai et al., 2011; Zhang et al., 2014), users retrieve historical questions which precisely match their questions (single sentence) semantically equivalent or relevant.

Neural network based architectures, especially CNN (Yin et al., 2016), LSTM (Mueller and Thyagarajan, 2016), RNN encoder-decoder (Kiros et al., 2015), etc. have shown success in similarity learning

ID	Topic Words (Top 10)
#83	pulsation, frequency, low, load, high, pulsations, increase, narrow, XXXX, mw
#7	trip, turbine, vibration, gas, alarm, gt, time, tripped, pressure, load
#30	start, flame, unit, turbine, combustion, steam, temperature, compressor, XXXX, detector
#16	oil, XXXX, XXXX, pressure, kpa, dp, level, high, mbar, alarm
#19	valve, XXXX, fuel, valves, gas, bypass, check, control, XXXX, XXXX

Table 8: Topics Identifier and words captured by DocNADE

task in Siamese framework (Mueller and Thyagarajan, 2016; Chopra et al., 2005). These models are adapted to similarity learning in sentence pairs using complex learners. Wieting et al. (2016) observed that word vector averaging and LSTM for similarity learning perform better in short and long text pairs, respectively. Our learning objective exploits the multi-channel representations of short and longer texts and compute cross-level similarities in different components of the query and tickets pairs. Instead of learning similarity in a single sentence pair, we propose a novel task and neural architecture for asymmetric textual similarities. To our knowledge, this is the first advancement of Siamese architecture towards multi-and-cross level similarity learning in asymmetric text pairs with an industrial application.

6 Conclusion and Discussion

We have demonstrated deep learning application in STS and IR tasks for an industrial ticketing system. The results indicate that the proposed LSTM is capable of modeling complex semantics by explicit guided representations and does not rely on hand-crafted linguistic features, therefore being generally applicable to any domain. We have showed improved similarity and retrieval via the proposed multi-and-cross-level Replicated Siamese architecture, leading to relevant recommendations especially in industrial use-case. As far we we know, this is the first advancement of Siamese architecture for similarity learning and retrieval in asymmetric text pairs with an industrial application.

We address the challenges in a real-world industrial application of ticketing system. Industrial assets like power plants, production lines, turbines, etc. need to be serviced well because an unplanned outage always leads to significant financial loss. It is an established process in industry to report issues (via query) i.e. symptoms which hint at an operational anomaly to the service provider. This reporting usually leads to textual descriptions of the issue in a ticketing system. The issue is then investigated by service experts who evaluate recommended actions or solutions to the reported issue. The recommended actions or solutions are usually attached to the reported issues and form a valuable knowledge base on how to resolve issues. Since industrial assets tend to be similar over the various installations and since they don't change quickly it is expected that the issues occurring over the various installations may be recurring. Therefore, if for a new issue similar old issues could be easily found this would enable service experts to speed up the evaluation of recommended actions or solutions to the reported issue. The chosen approach is to evaluate the pairwise semantic similarity of the issues describing texts.

We have compared unsupervised and supervised approach for both similarity learning and retrieval tasks, where the supervised approach leads the other. However, we foresee significant gains with the larger amount of similarity data as the amount of labeled similarity data grows and the continuous feedback is incorporated for optimization within the industrial domain, where quality results are desired. In future work, we would also like to investigate attention (Bahdanau et al., 2014) mechanism and dependency (Socher et al., 2012; Gupta et al., 2018b) structures in computing tickets' representation.

Acknowledgements

We thank our colleagues Mark Buckley, Stefan Langer, Subburam Rajaram and Ulli Waltinger, and anonymous reviewers for their review comments. This research was supported by Bundeswirtschaftsministerium (bmwi.de), grant 01MD15010A (Smart Data Web) at Siemens AG- CT Machine Intelligence, Munich Germany.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representation*, Alberta, Canada.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao. 2011. Learning the latent topics for question retrieval in community qa. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 273–281, Chiang Mai, Thailand. Association of Computational Linguistics.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 539–546, San Diego, CA, USA. IEEE.

Pankaj Gupta and Bernt Andrassy. 2018. Device and method for natural language processing. US Patent 2018-0,157,643.

Pankaj Gupta, Vijay Shankar Pendluri, and Ishant Vats. 2011. Summarizing text by ranking text units according to shallow linguistic features. Seoul, South Korea. IEEE.

Pankaj Gupta, Thomas Runkler, Heike Adel, Bernt Andrassy, Hans-Georg Zimmermann, and Hinrich Schütze. 2015a. Deep learning methods for the extraction of relations in natural language text. Technical report, Technical University of Munich, Germany.

Pankaj Gupta, Thomas Runkler, and Bernt Andrassy. 2015b. Keyword learning for classifying requirements in tender documents. Technical report, Technical University of Munich, Germany.

Pankaj Gupta, Udhayaraj Sivalingam, Sebastian Pölsterl, and Nassir Navab. 2015c. Identifying patients with diabetes using discriminative restricted boltzmann machines. Technical report, Technical University of Munich, Germany.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan.

Pankaj Gupta, Florian Buettner, and Hinrich Schütze. 2018a. Document informed neural autoregressive topic models. Researchgate preprint doi: 10.13140/RG.2.2.12322.73925.

Pankaj Gupta, Subburam Rajaram, Bernt Andrassy, Thomas Runkler, and Hinrich Schütze. 2018b. Neural relation extraction within and across sentence boundaries. Researchgate preprint doi: 10.13140/RG.2.2.16517.04327.

Pankaj Gupta, Subburam Rajaram, Hinrich Schütze, and Bernt Andrassy. 2018c. Deep temporal-recurrent-replicated-softmax for topical trends over time. 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 1079–1089, New Orleans, USA. Association of Computational Linguistics.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2009. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pages 1607–1614, Vancouver, Canada.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, Montreal, Canada.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270. Association for Computational Linguistics.

Hugo Larochelle and Stanislas Lauly. 2012. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pages 1607–1614, Lake Tahoe, USA. Curran Associates, Inc.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, page 3.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, Lake Tahoe, USA.

G.A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):3941.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *the thirtieth AAAI conference on Artificial Intelligence*, volume 16, pages 2786–2792, Phoenix, Arizona USA.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 1310–1318, Atlanta, USA.

Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine learning*, pages 791–798, Oregon, USA. Association for Computing Machinery.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211, Jeju Island, Korea. Association for Computational Linguistics.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016a. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539, San Diego, California USA. Association for Computational Linguistics.

Ngoc Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schütze. 2016b. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *Proceedings of the Acoustics, Speech and Signal Processing (ICASSP)*, pages 6060–6064, Shanghai, China. IEEE.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *Proceedings of the International Conference on Learning Representations*, San Juan, Puerto Rico.

Wenpeng Yin, Hinrich Schuetze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272.

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

Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. 2014. Question retrieval with high quality answers in community question answering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 371–380, Shanghai, China. Association for Computing Machinery.

Jiang Zhao, Tiantian Zhu, and Man Lan. 2014. Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 271–277, Dublin, Ireland.

Le Zhao. 2012. Modeling and solving term mismatch for full-text retireval. *ACM SIGIR*, pages 117–118.

Yin Zheng, Yu-Jin Zhang, and Hugo Larochelle. 2016. A deep and autoregressive approach for topic modeling of multimodal data. *IEEE transactions on pattern analysis and machine intelligence*, 38(6):1056–1069.