




Simultaneous Contextualization and Interpretation with Keyword Awareness

Tepei Yoshino^(✉), Shoya Matsumori, Yosuke Fukuchi, and Michita Imai

Keio University, Yokohama, Japan
yoshino@ailab.ics.keio.ac.jp

Abstract. Most natural-language-processing methods are designed for estimating context given an entire set of sentences at once. However, dialogue is incremental in nature. SCAIN (Simultaneous Contextualization and Interpretation) is an algorithm for incremental dialogue processing. Along with the progress of the dialogue, it can solve the interdependence problem in which the interpretation of words depends on the context, and the context is determined by the interpreted words. However, SCAIN cannot process texts that contain more words insignificant to context estimation such as in longer texts. We propose SCAIN with keyword extraction (SCAIN/KE), which extracts keywords that contribute to context estimation and eliminates the effect of insignificant words so that it can process longer texts. In the case study, SCAIN/KE updates context and interpretation better than SCAIN and obtains the keywords that contribute to context estimation better than other statistical methods. In the experiments, we evaluated SCAIN/KE on solving the ambiguity of polysemous words using the Wikipedia disambiguation pages. The results indicate that SCAIN/KE is more accurate than SCAIN.

Keywords: Dialogue context · Polysemy · Keyword extraction · SCAIN · SLAM

1 Introduction

A word can have multiple meanings, and such words are known as polysemous words. Previous studies proposed methods of processing polysemous words in natural-language processing [6, 12]. A dialogue system needs to process a polysemy of word meanings and be able to identify what a word means in a conversation to interpret the speaker's intent. Context, which is composed of previous utterances, contains critical information that contributes to resolving the ambiguity of word meaning. However, most current dialogue systems only handle single-round conversation, such as a query-response pair, or predetermine the domain of a conversation and cannot take into account the context. Computational methods for context-aware word interpretation will help dialogue systems properly remove ambiguity in interpreting utterances.

There are many challenges in designing such a method. One important problem is the interdependence between context and word meaning: the interpretation of a word depends on the context, however, the context is determined by the interpreted words. Moreover, in a conversation, such interdependence needs to be processed sequentially. Even when the context of the dialogue is still unclear, it is necessary to interpret the meaning of an utterance and infer the context from undefined words to keep the conversation going. The dialogue system may need to withhold the interpretation of words to continue a dialogue and revise word interpretation in response to subsequent utterances. The system must carry out the following two processes simultaneously to sequentially determine the meaning of words and context in a dialogue. The first is estimating and retaining possible contexts under a certain interpretation of utterances. The second is continuously evaluating the context candidates on the basis of the interpretations of past utterances.

SCAIN [11] is an algorithm for identifying the meaning of words depending on contexts and estimating contexts depending on utterances in an incremental manner. SCAIN is based on FastSLAM [4], which is an algorithm designed for mobile robots to statistically resolve the interdependence between the robot's self-position and a map. SCAIN replaces self-position with a context and the map with a word-interpretation space to apply FastSLAM to the interdependence between context and interpretation in a sequential dialogue. In particular, the Kalman filter and a particle filter are the primary mechanisms recruited from FastSLAM. The particle filter holds multiple contexts at the same time, and word ambiguities can be clarified by selecting the interpretation with the more likely context.

However, SCAIN is not ideal with respect to processing long sentences. One of the reasons is that it uses a simple average of word vectors in estimating a context. Because of this, insignificant words that should not contribute to context estimation adversely affect the calculation of context likelihood. The more word vectors entered, the more their mean vectors converge to the center of the word-embedding space. To avoid this, it is necessary to distinguish between the words that should contribute to the context and those that should not.

We propose SCAIN with keyword extraction (SCAIN/KE). We improved upon SCAIN by introducing the idea of keywords, which are useful in estimating context. SCAIN/KE selects keywords on the basis of the assumption that the vectors of important words in a dialogue are located around a context vector that represents the entire dialogue history. A keyword extraction algorithm uses SCAIN's function in which possible context candidates are estimated in a particle-wise manner. Because SCAIN holds various possible contexts as particles, it can infer possible keywords on the basis of their possible contexts. With keyword extraction, we can reduce the effect of insignificant words and obtain more accurate context and word interpretation.

We conducted an experiment involving the Wikipedia disambiguation pages to carry out a polysemy disambiguation task and revealed that SCAIN/KE could disambiguate polysemous words more successfully than SCAIN, which indicates that by introducing the concept of keywords, SCAIN/KE can effectively resolve the problem of the interdependence between context estimation and word interpretation.

The remainder of this paper is structured as follows. In Sect. 2, we present related work regarding dialogue-context estimation and polysemy resolution and explain their challenges. In Sect. 3, we describe SCAIN/KE and investigate its effectiveness with an example dialogue. In Sect. 4, we discuss an experiment we conducted involving the Wikipedia-based polysemy resolution task used in a previous study [11], which showed that SCAIN/KE has better interpretation performance than SCAIN. Finally, we conclude the paper in Sect. 5.

2 Related Work

2.1 Multi-sense Embedding

We focus on handling the ambiguity of word meanings in dialogue. Many studies proposed word-embedding techniques that take into account polysemy. The technique word2gauss [12] uses Gaussian distribution as a representation of a word to express the ambiguity of meaning. ELMo [6] obtains context-aware word representation by concatenating a context vector with an existing word vector. BERT [2] uses masked language modeling to obtain deep bidirectional context-aware representations.

However, word2gauss acquires only the semantic field to which the word is assigned from datasets and cannot take into account the context. ELMo and BERT can interpret words from context, however, their representation of a word meaning is deterministic for each input sentence, which is problematic for sequential dialogue processing. Word meaning cannot be estimated deterministically. That is, we cannot necessarily determine the meaning of a word when it appears, and it is often the case that what a speaker intends to convey with a word is gradually clarified as the dialogue progresses. A dialogue system should retain multiple interpretations of words inferred from the current dialogue history then revise them sequentially.

2.2 Dialogue-Context Estimation

For estimating a word’s meaning, it is important to infer its context, especially the long-term context, which is built from the dialogue history. HRED [9] infers a context vector using the *encoder* recurrent neural network (RNN), which embeds an utterance to the distributed representation space, and the *context* RNN which generates a context vector from the outputs of the encoder RNN. MemN2N [10] stores multiple sentence vectors in an external memory and uses them to generate a context vector using an attention mechanism [1]. Both methods can generate dialogue context representation by taking into account dialogue history, but they are not applicable to sequential dialogue processing because it is not necessarily possible to identify the exact meaning of an utterance when it is given. The interpretation of an utterance is gradually updated along with the progress in the dialogue, as we discussed in Sect. 2.1; thus, the inferred context should also be updated accordingly.

2.3 SCAIN

SCAIN [11] is an algorithm that sequentially infers context and word interpretation and based on FastSLAM [4]. SCAIN can solve the problem of the interdependence between context and interpretation, i.e., the context determines the interpretation of a word, and the context is determined by a set of words with a fixed interpretation.

In SCAIN, context x is represented by the locus of a point in a word-embedding space. The word-interpretation space m is represented by pairs of a word label and Gaussian distribution with mean μ and covariance matrix Σ in the same word-embedding space as the context vector. The m indicates possible interpretations of all words that appeared in a dialogue history. In SCAIN, the combination of x and m is represented as a particle; a single particle represents an instance of utterance comprehension, which consists of an estimated x and set of words m interpreted in that context. This enables SCAIN to sequentially interpret ambiguous words on the basis of the context while simultaneously inferring the context on the basis of the word interpretation.

However, SCAIN does not work when processing long utterance texts because of how it infers context. With SCAIN, it is assumed that the context can be simply calculated by an average of word vectors appearing in a sentence. However, all the words that appeared in an utterance are not necessarily important to infer its context. A sentence usually has words that are closely related to its context and other insignificant words. Therefore, we need to consider the connection between each word in a sentence and its context.

3 SCAIN/KE

SCAIN/KE extracts the words that represent the entire text, or keywords, by taking into account the distance between a word and its context. These keywords are used in calculating context likelihood and expected to improve context estimation. By using SCAIN's feature of holding possible sets of context and interpretation of words, SCAIN/KE can taking into account keywords even when the context and meaning of words in a dialogue remain ambiguous. Although some particles may have wrong contexts and keywords, as the dialogue continues and the context becomes clearer, appropriate particles are selected and the correct contexts and keywords gradually become dominant. SCAIN/KE consists of the following three steps.

3.1 Contextualization

SCAIN/KE updates x by using Eq. (1):

$$x_{t+1}^k = (1 - \lambda_u) x_t^k + \lambda_u v_{ut} + \sigma_u, \quad (1)$$

$$v_{ut} = \frac{1}{N} \sum_i^N w_i \cos(x_t^k, w_i). \quad (2)$$

where u is an utterance, λ_u is the learning rate, σ_u is the Gaussian noise corresponding to the update error of a context, and N is the number of words that appeared in utterance u . In Eq. (2), a word vector w_i is weighted by the word importance calculated from cosine similarity. This is a unique step of SCAIN/KE; SCAIN simply calculates v_{ut} as the average of w_i . Since this weight increases as the distance between a word and context decreases, the movement of the context vector is small if a word appears around the context. With this implementation, we can prevent particles that had acquired the correct context from moving to the center of the word-embedding space due to insignificant words.

3.2 Interpretation

After updating the context vectors, SCAIN/KE recalculates the interpretation distributions of words uttered by the user in each particle. This step conforms to SCAIN. First, each observation word vector z_i is calculated from pre-trained word vector w_i with $z_i = (1 - \alpha)w_i + \alpha x_{t+1}^k$, where α is the parameter indicating how much z_i is drawn to the context vector. This is explained as observation noise. To minimize this noise, the Kalman filter is applied to the interpretation distribution. These processes give each word distribution (μ_i, Σ_i) .

3.3 Resampling

For each particle, the likelihood w is calculated from the pair of word distributions and estimated context vector in Eq. (3). Particles are resampled with reference to each particle w . To mitigate interference from insignificant words, the w s of the particles that detected keywords are summed with parameter λ_D .

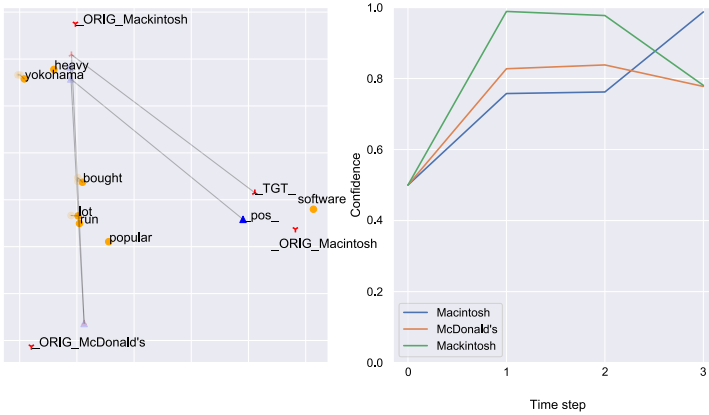
$$w = -\log \left(\frac{1}{N} \sum_{i=1}^N \eta(l_i) D_M(\langle \mu_i, \Sigma_i \rangle, x) + \epsilon \right) - \lambda_D d_x, \quad (3)$$

$$d_x = \min(D_M(\mu, \Sigma)),$$

where D_M is the Mahalanobis distance representing the distance between the distribution and vector, ϵ is a minimal value to avoid division by zero, η is an attenuation term to account for the time when the interpretation is updated, l is a word label, N is the total number of uttered words, and d_x is the minimum distance between context and words. The second term in Eq. (3) implies a preference for the particles that have detected keywords, and it is one of the unique points of SCAIN/KE. The reason the weights of the particles nearing the uttered words are added under λ_D is as follows. If input sentences contain many insignificant words, the sums of the distances between all uttered words and each context vector become almost uniform. By enabling preferential treatment of the second term under a constraint λ_D , the weight of the particle is a bit dispersed even if the computation of the first term is smoothed.

Table 1. An example dialogue.

Speaker	Utterance	Time step
Human	I bought a mac	0
Agent	Where did you buy it?	1
Human	I bought it in Yokohama	1
Agent	How was it?	2
Human	There were a lot of people	2
Agent	It seems popular	3
Human	Yes, it is popular It can run heavy software	3

**Fig. 1.** (left) Visualized map of context and word vectors. (right) Confidence of interpretation of *mac* along time steps (SCAIN/KE). (Color figure online)

3.4 Case Study on SCAIN/KE

Case Study on Context and Word Sense. To examine SCAIN/KE in terms of sequential dialogue processing, we conducted a disambiguation task in a dialogue. We prepared an example dialogue in which the meaning of the polyseme *mac* is gradually revealed. We observed the transition of estimated context and word interpretation with SCAIN/KE. We also compared SCAIN/KE with SCAIN using the same dialogue. We defined the *mac* word vectors as one of the following three: *McDonald's* (hamburger), *Mackintosh* (coat), or *Macintosh* (computer). The word vectors for them were obtained from their respective Wikipedia pages. We used the pre-trained GloVe [5] 100-dimension word embeddings to define the original vectors of the words. The input sentences are listed in Table 1. The dialogue is a conversation between a person and an agent that infers the meaning of the word *mac*.

Figures 1 and 2 show the transition of the context and confidence of interpretations of *mac* with SCAIN/KE and SCAIN, respectively.

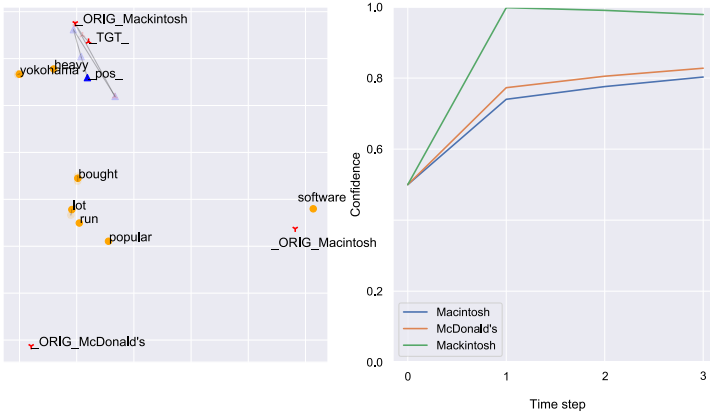


Fig. 2. (left) Visualized map of context and word vectors. (right) Confidence of interpretation of *mac* along time steps (SCAIN). (Color figure online)

The left sides of Figs. 1 and 2 show the word-embedding space that SCAIN/KE and SCAIN maintain in their particles. It is visualized by applying a principal component analysis to disperse three *mac* vectors. For visualization, we display only the context and word interpretations of the particle with the highest weight at each time step. In these figures, `_pos_`, represented as a blue triangle, is the position of the estimated context, `_TGT_` with a red triangle is the mean of the word distribution labeled *mac*, and `_ORIG_` with a red inverted triangle is a pretrained word vector of each interpretation of *mac*. Other word labels are the mean of their word distributions. The right sides of Figs. 1 and 2 show the confidence of interpretation of *mac* along time steps. The confidence was calculated from the cosine similarity between the *mac* vector of the particle with the highest weight and each candidate interpretation vector. SCAIN/KE interpreted *mac* as a computer with the highest (0.99) confidence at time step 3.

As shown on the left of Fig. 1, the context vector moved noticeably at time step 3 and reached *Macintosh*. This move occurred because SCAIN/KE estimated *software* as a keyword and recognized that the utterance was about computers. As shown on the right of Fig. 1, at time step 3 the confidence of *Macintosh* increased and decreased for the others. This is because the example dialogue does not provide any useful information on the interpretation of *mac* until time step 2, and it is not until time step 3 that we infer that it is a *Macintosh*.

As shown on the left of Fig. 2, in SCAIN, which does not take into account keywords, the context vector remained stuck to *Mackintosh* (coat). These results indicate that keyword extraction contributed to correct context inference.

Case Study on Keyword Extraction. To investigate the keyword extraction with SCAIN/KE, we compared the results of keyword extraction with three other methods: TFIDF [8], TextRank [3], and RAKE [7]. Because TFIDF requires other general documents, we used the NPS Chat Corpus, which consists of more

Table 2. Comparison of keyword extraction (with relative word-importance values).

SCAIN/KE	TFIDF	TextRank	RAKE
Computer (0.164)	Delivering (0.158)	A powerful computer (0.371)	Play latest games (0.529)
Better (0.126)	Powerful (0.158)	The latest games (0.368)	Powerful computer (0.235)
Latest (0.118)	Latest (0.137)	The power (0.262)	Power (0.059)
Games (0.115)	Games (0.137)	You (0.000)	Need (0.059)
Need (0.108)	Power (0.137)	–	Delivering (0.059)
Power (0.107)	Computer (0.088)	–	Better (0.059)
Play (0.095)	Play (0.068)	–	–
Powerful (0.088)	Better (0.060)	–	–
Delivering (0.079)	Need (0.057)	–	–

Table 3. Cosine similarities with *Macintosh*.

Word	Cosine similarity
Computer	0.756
Latest	0.444
Better	0.423
Games	0.415
Power	0.337
Need	0.328
Play	0.266
Powerful	0.265
Delivering	0.243

than 10,000 posts from chat rooms. We input the example sentence “If you will play the latest games, a powerful computer will be better for delivering the power you need.” after the dialogue shown in Table 1 and compared the importance rate of each word. The results are listed in Table 2.

SCAIN/KE estimated that *computer* is important while *delivering* is not, whereas TFIDF inferred *delivering* is the keyword. TextRank assigned similar importance to *a powerful computer*, *the latest games*, and *the power*, respectively. RAKE recognized *play latest games* and *powerful computer* as idioms and regarded them as important.

The example sentence was talking about a Macintosh computer. Based on Eqs. (1) and (2), it is helpful to extract words near Macintosh as keywords to properly infer the context of this example sentence. Table 3 shows the cosine similarity between each word except for stopwords in the example sentence and

Macintosh. In accordance with Table 3, the word with the highest cosine similarity with *Macintosh* was *computer*. SCAIN/KE estimated *computer* to be significantly more important than the other words. Therefore, we can expect that the proposed keyword extraction algorithm enables SCAIN/KE to infer dialogue context more accurately than the other methods.

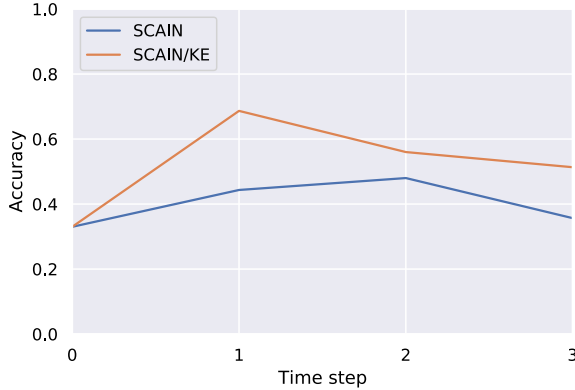


Fig. 3. Accuracy of polysemy resolutions.

4 Evaluation

4.1 Method

In a similar manner as in a previous study [11], we conducted an experiment on polysemy resolution and compared the results of SCAIN/KE with those of SCAIN. The experiment was conducted using polysemous words from Wikipedia; the disambiguation page of Wikipedia provides ambiguous word labels and descriptions of the label’s interpretation candidates of the polysemous word. We randomly selected 300 disambiguation pages and extracted three candidates as possible interpretations for each page. The procedure of the experiment was as follows. First, we chose a specific topic from a disambiguation page. For each topic, we input the label of the polysemous word as a first utterance into both SCAIN and SCAIN/KE. We then input the description sentences as the following utterances. We evaluated how the meanings of the polysemous words were updated as the time steps progressed to consider the accuracy of SCAIN/KE’s sequential dialogue processing. For each sentence entered, we calculated the cosine similarity between the updated polysemous word vectors and those of each correct answer. We investigated whether a candidate with the highest similarity in the particle with the highest likelihood was the correct interpretation.

4.2 Results

Figure 3 shows the results of this experiment. The horizontal axis is the number of sentences entered and the vertical axis is accuracy. At time step 0, we input

only polysemous words, and from time step 1, we input one sentence per one time step. SCAIN/KE estimated the meanings of polysemous words with higher accuracy than SCAIN. In particular, SCAIN/KE had an accuracy of 0.69 in time step 1, while SCAIN had an accuracy of 0.44, indicating that SCAIN/KE could successfully update the meaning of the previous utterance when the next utterance was entered. These results suggest that, by introducing the concept of keywords, SCAIN/KE is better at solving the interdependence problem of a dialogue’s context and word interpretation than SCAIN. There are possible reasons the accuracy rate did not increase as the dialogue progressed. First, some tasks generated from Wikipedia were too difficult to solve because some interpretation candidates on Wikipedia’s disambiguation page were very similar to each other. Second, because Wikipedia articles are often written to reveal the topic in the first sentence, we could not fully simulate the dialogue as it gradually became clearer as the time steps progressed. A dataset that gradually reveals polysemous words as a dialogue progresses would have yielded more practical results.

5 Conclusions

We proposed SCAIN/KE, an algorithm for sequentially interpreting utterances under the problem of interdependence between context and word meaning. SCAIN/KE exploits the idea of keywords to improve the inference of context. We conducted an experiment to compare SCAIN/KE with SCAIN in a word-sense disambiguation task. The results indicate that, by using SCAIN/KE, we could estimate both the context and interpretation of utterance texts better in processing ongoing dialogue.

Acknowledgements. This work was supported by JST CREST Grant Number JPMJCR19A1, Japan.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, January 2015
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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), Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics, June 2019
3. Mihalcea, R., Tarau, P.: TextRank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004)
4. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., et al.: FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: AAAI/IAAI, p. 593598 (2002)

5. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
6. Peters, M., et al.: 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), New Orleans, Louisiana, pp. 2227–2237. Association for Computational Linguistics, June 2018. <https://doi.org/10.18653/v1/N18-1202>, <https://www.aclweb.org/anthology/N18-1202>
7. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual documents. *Text Min. Appl. Theor.* **1**, 1–20 (2010)
8. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.* **24**(5), 513–523 (1988)
9. Serban, I.V., et al.: A hierarchical latent variable encoder-decoder model for generating dialogues. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 3295–3301 (2017)
10. Sukhbaatar, S., Weston, J., Fergus, R., et al.: End-to-end memory networks. In: Advances in Neural Information Processing Systems, pp. 2440–2448 (2015)
11. Takimoto, Y., Fukuchi, Y., Matsumori, S., Imai, M.: Slam-inspired simultaneous contextualization and interpreting for incremental conversation sentences. arXiv preprint [arXiv:2005.14662](https://arxiv.org/abs/2005.14662) (2020)
12. Vilnis, L., McCallum, A.: Word representations via gaussian embedding. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015)