

Energy-based Self-attentive Learning of Abstractive Communities for Spoken Language Understanding

Guokan Shang^{1,2}, Antoine J.-P. Tixier¹,
Michalis Vazirgiannis^{1,3}, Jean-Pierre Lorré²

¹École Polytechnique, ²Linagora, ³AUEB

Abstract

Abstractive community detection is an important spoken language understanding task, whose goal is to group utterances in a conversation according to whether they can be jointly summarized by a common abstractive sentence. This paper provides a novel approach to this task. We first introduce a neural contextual utterance encoder featuring three types of self-attention mechanisms. We then train it using the siamese and triplet energy-based meta-architectures. Experiments on the AMI corpus show that our system outperforms multiple energy-based and non-energy based baselines from the state-of-the-art. Code and data are publicly available¹.

1 Introduction

Today, large amounts of digital text are generated by spoken or written conversations, let them be human-human (customer service, multi-party meetings) or human-machine (chatbots, virtual assistants). Such text comes in the form of transcriptions. A transcription is a list of time-ordered text fragments called *utterances*. Abstractive summarization of conversations is an open problem in NLP. Previous work (Mehdad et al., 2013; Oya et al., 2014; Banerjee et al., 2015; Shang et al., 2018) decomposes this task into two subtasks a and b as shown in Fig. 1.

Subtask a, or *Abstractive Community Detection* (ACD), is the focus of this paper. It consists in grouping utterances according to whether they can be jointly summarized by a common abstractive sentence (Murray et al., 2012). Such groups of utterances are called *abstractive communities*. Once they are obtained, an abstractive sentence is generated for each group (subtask b), thus forming the final summary. ACD includes, but is a more

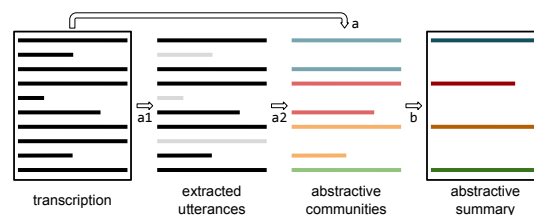


Figure 1: Abstractive summarization of conversations.

general problem than, topic clustering. Indeed, as shown in Fig. 2, communities should capture more complex relationship than simple semantic similarity. Also, two utterances may be part of the same community even if they are not close to each other in the transcription. Finally, a given utterance may belong to more than one community, which results in overlapping groupings (e.g., A and D in Fig. 2), or be a singleton community (B in Fig. 2).

In this paper, we depart from previous work and argue that the ACD subtask should be broken down into two steps, a1 and a2 in Fig. 1. That is, summary-worthy utterances should first be extracted from the transcription (a1), and then, grouped into abstractive communities (a2). This process is more consistent with the way humans treat the summarization task. E.g., during the creation of the AMI corpus (McCowan et al., 2005), annotators were first asked to extract summary-worthy utterances from the transcription, and then to link the selected utterances to the sentences in the abstractive summary (links in Fig. 2), i.e., create communities. Abstractive summaries comprise four sections: ABSTRACT, ACTIONS, PROBLEMS, and DECISIONS. Step a1 plays an important filtering role, since in practice, only a small part of the original utterances are used to construct the abstractive communities (17% on average for AMI). However, this step is closely related to *extractive summarization*, which has been extensively studied in the conversational domain (Murray et al.,

¹https://bitbucket.org/guokan_shang/abscomm

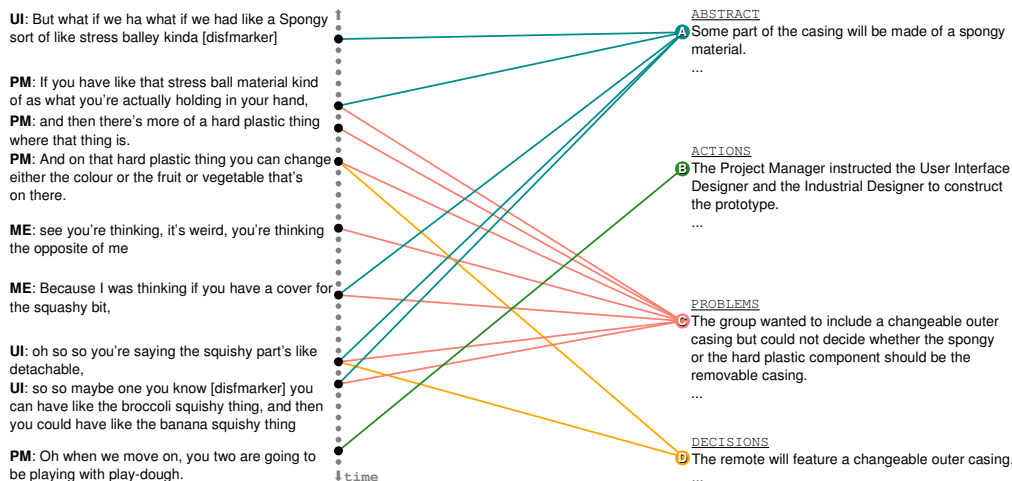


Figure 2: Example of ground truth human annotations from the ES2011c AMI meeting. Successive grey nodes on the left denote utterances in the transcription. Black nodes correspond to the utterances judged important. Sentences (e.g., A, B, C, D) from the abstractive summary are shown on the right. All utterances linked to the same abstractive sentence form one community.

2005; Garg et al., 2009; Tixier et al., 2017).

Rather, we focus in this paper on the rarely explored *a2 utterance clustering* step, which we think is an important spoken language understanding problem, as it plays a crucial role of bridge between two major types of summaries: extractive and abstractive.

2 Departure from previous work

Prior work performed ACD either in a supervised (Murray et al., 2012; Mehdad et al., 2013) or unsupervised way (Oya et al., 2014; Banerjee et al., 2015; Singla et al., 2017; Shang et al., 2018).

In the supervised case, Murray et al. (2012) train a logistic regression classifier with handcrafted features to predict extractive-abstractive links, then build an utterance graph whose edges represent the binary predictions of the classifier, and finally apply an overlapping community detection algorithm to the graph. Mehdad et al. (2013) add to the previous approach by building an entailment graph for each community, where edges are entailment relations between utterances, predicted by a SVM classifier trained with handcrafted features on an external dataset. The entailment graph allows less informative utterances to be eliminated from each community.

On the other hand, unsupervised approaches to ACD do not make use of extractive-abstractive links. Oya et al. (2014); Banerjee et al. (2015); Singla et al. (2017) assume that disjoint topic segments (Galley et al., 2003) align with abstractive communities, while Shang et al. (2018) use the classical vector space representation with TF-

IDF weights, and apply k -means to the LSA-compressed utterance-term matrix.

To sum up, prior ACD methods either train multiple models on different labeled datasets and heavily rely on handcrafted features, or are incapable of capturing the complicated structure of abstractive communities described in the introduction.

Motivated by the recent success of energy-based approaches to similarity learning tasks such as face verification (Schroff et al., 2015) and sentence matching (Mueller and Thyagarajan, 2016), we introduce in this paper a novel utterance encoder, and train it within the siamese (Chopra et al., 2005) and triplet (Hoffer and Ailon, 2015) energy-based meta-architectures. Our final network is able to accurately capture the complexity of abstractive community structure, while at the same time, it is trainable in an end-to-end fashion without the need for human intervention and handcrafted features. Our contributions are threefold: (1) we formalize ACD, a crucial subtask for abstractive summarization of conversations, and publicly release a version of the AMI corpus preprocessed for this subtask, to foster research on this topic, (2) we propose one of the first applications of energy-based learning to spoken language understanding, and (3) we introduce a novel utterance encoder featuring three types of self-attention mechanisms and taking contextual and temporal information into account.

3 Energy framework

Energy-Based Modeling (EBM) (LeCun and Huang, 2005; Lecun et al., 2006) is a unified framework that can be applied to many machine learning

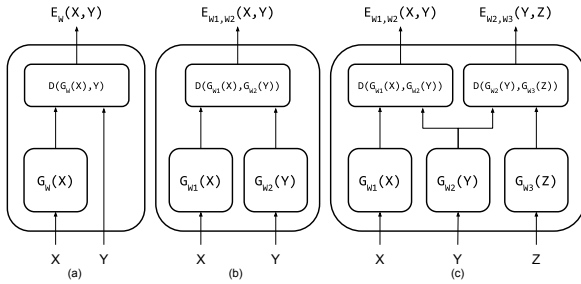


Figure 3: Three EBM architectures. When all G s and W s are equal, (b) and (c) correspond to the siamese/triplet cases.

problems. In EBM, an energy function assigns a scalar called *energy* to each pair of random variables (X, Y) . The energy can be interpreted as the incompatibility between the values of X and Y . Training consists in finding the parameters W^* of the energy function E_W that, for all (X^i, Y^i) in the training set \mathcal{S} of size P , assign low energy to compatible (correct) combinations and high energy to all other incompatible (incorrect) ones. This is done by minimizing a *loss functional*² \mathcal{L} :

$$W^* = \arg \min_{W \in \mathcal{W}} \mathcal{L}(E_W(X, Y), \mathcal{S}) \quad (1)$$

For a given X , prediction consists in finding the value of Y that minimizes the energy.

3.1 Single architecture

In the EBM framework, a regression problem can be formulated as shown in Fig. 3a, where the input X is passed through a regressor model G_W and the scalar output is compared to the desired output Y with a dissimilarity measure D such as the squared error. Here, the energy function is the loss functional to be minimized.

$$\mathcal{L} = \frac{1}{P} \sum_{i=1}^P E_W(X^i, Y^i) = \frac{1}{P} \sum_{i=1}^P \|G_W(X^i) - Y^i\|^2 \quad (2)$$

3.2 Siamese architecture

In the regression problem previously described, the dependence between X and Y is expressed by a direct mapping $Y = f(X)$, and there is a single best Y^* for every X . However, when X and Y are not in a predictor/predictand relationship but are exchangeable instances of the same family of objects, there is no such mapping. E.g., in paraphrase identification, a sentence may be similar

²the *loss functional* is passed the output of the energy function, unlike a *loss function* which is directly fed the output of the model.

to many other ones, or, in language modeling, a given n -gram may be likely to be followed by many different words.

Thereby, [Lecun et al. \(2006\)](#) introduced EBM for *implicit regression* or *constraint satisfaction* (see Fig. 3b), in which a constraint that X and Y must satisfy is defined, and the energy function measures the extent to which that constraint is violated:

$$E_{W_1, W_2}(X, Y) = D(G_{W_1}(X), G_{W_2}(Y)) \quad (3)$$

where G_{W_2} and G_{W_1} are two functions parameterized by W_1 and W_2 . When $G_{W_1} = G_{W_2}$ and $W_1 = W_2$, we obtain the well-known siamese architecture ([Bromley et al., 1994](#); [Chopra et al., 2005](#)), which has been applied with success to many tasks, including sentence similarity ([Mueller and Thyagarajan, 2016](#)).

Here, the constraint is determined by a collection-level set of binary labels $\{C^i\}_{i=1}^P$. $C^i = 0$ indicates that (X^i, Y^i) is a *genuine* pair (e.g., two paraphrases), while $C^i = 1$ indicates that (X^i, Y^i) is an *impostor* pair (e.g., two sentences with different meanings).

The function G_W projects objects into an embedding space such that the defined dissimilarity measure D (e.g., Euclidean distance) in that space reflects the notion of dissimilarity in the input space. Thus, the energy function can be seen as a metric to be learned.

We experiment with deep neural network encoders as G_W , and, following ([Mueller and Thyagarajan, 2016](#)), we adopt the exponential negative Manhattan distance as dissimilarity measure and the mean squared error as loss functional:

$$E_W(X, Y) = 1 - \exp(-\|G_W(X) - G_W(Y)\|_1) \quad (4)$$

$$\mathcal{L} = \frac{1}{P} \sum_{i=1}^P \|E_W(X^i, Y^i) - C^i\|^2 \quad (5)$$

3.3 Triplet architecture

The triplet architecture ([Schroff et al., 2015](#); [Hoffer and Ailon, 2015](#); [Wang et al., 2014](#)), as can be seen in Fig. 3c, is a direct extension of the siamese architecture that takes as input a triplet (X, Y, Z) in lieu of a pair (X, Y) . X , Y , and Z are referred to as the *positive*, *anchor*, and *negative* objects, respectively. X and Y are similar, while both being dissimilar to Z . Learning consists in jointly minimizing the positive-anchor energy $E_W(X^i, Y^i)$ while maximizing the anchor-negative energy $E_W(Y^i, Z^i)$.

Here, we use the *softmax triplet loss* ([Hoffer and Ailon, 2015](#)) as our loss functional:

$$\mathcal{L} = \frac{1}{2P} \sum_{i=1}^P (\|ne^+ - 0\|^2 + \|ne^- - 1\|^2) \quad (6)$$

$$ne^+ = \frac{e^{E_W(X^i, Y^i)}}{e^{E_W(X^i, Y^i)} + e^{E_W(Y^i, Z^i)}} \quad (7)$$

$$ne^- = \frac{e^{E_W(Y^i, Z^i)}}{e^{E_W(X^i, Y^i)} + e^{E_W(Y^i, Z^i)}} \quad (8)$$

where ne stands for normalized energy, and the dissimilarity measure is the Euclidean distance, i.e., $E_W(X^i, Y^i) = \|G_W(X^i) - G_W(Y^i)\|_2$. Essentially, the softmax triplet loss is the mean squared error between the normalized energy vector $[ne^+, ne^-]$ and $[0, 1]$. We justify our choice of loss functionals in App. F.

3.4 Sampling procedures

We sample tuples from the ground truth abstractive communities to train our utterance encoder G_W under the siamese and triplet meta-architectures as follows.

Pair sampling. All utterances belonging to the same community are paired as genuine pairs, while impostor pairs are any two utterances coming from different communities.

Triplet sampling. Utterances from the same community provide positive and anchor items, while the negative item is taken from any other community.

4 Proposed utterance encoder

Notation. The *time* t (as superscript) denotes the position of a given utterance in the conversation of length T , and the *position* i (as subscript) denotes the position of a token within a given utterance of length N . E.g., \mathbf{u}_1^t is the representation of the first token of \mathbf{U}^t , the t^{th} utterance in the transcription.

Word encoder. As shown in the upper right corner of Fig. 4, we obtain \mathbf{u}_i^t by concatenating the pre-trained vector of the corresponding token with the discourse features of \mathbf{U}^t (role, position and dialogue act), and passing the resulting vector to a dense layer.

Utterance encoder. As shown in the center of Fig. 4, we represent \mathbf{U}^t as a sequence of N d -dimensional token representations $\{\mathbf{u}_1^t, \dots, \mathbf{u}_N^t\}$. In addition, because there is a strong time dependence between utterances (see Fig. 2), we inform the model about the preceding and following utterances when encoding \mathbf{U}^t . To accomplish this, we prepend (resp. append) to \mathbf{U}^t a context vector containing information about the

previous (resp. next) utterances, finally obtaining $\mathbf{U}^t = \{\mathbf{u}_{\text{pre}}^t, \mathbf{u}_1^t, \dots, \mathbf{u}_N^t, \mathbf{u}_{\text{post}}^t\} \in \mathbb{R}^{(N+2) \times d}$. We then use a non-stacked bidirectional Recurrent Neural Network (RNN) with Gated Recurrent Units (GRU) (Cho et al., 2014) to transform \mathbf{U}^t into a sequence of annotations $\mathbf{H}^t \in \mathbb{R}^{(N+2) \times 2d}$.

In practice, the pre and post-context vectors indirectly initialize the left-to-right and right-to-left RNNs with information about the utterances preceding and following \mathbf{U}^t . This is similar in spirit to the warm-start method of Wang et al. (2017), that directly initializes the hidden states of the RNNs with the context vectors.

Self-attention. We then pass the annotations \mathbf{H}^t to a self-attention mechanism (Vaswani et al., 2017; Lin et al., 2017). More precisely, \mathbf{H}^t go through a dense layer and the output is compared via dot product with a trainable vector \mathbf{u}_γ , initialized randomly. Then, a probability distribution over the $N + 2$ tokens in \mathbf{U}^t is obtained via a softmax:

$$\gamma^t = \text{softmax}(\mathbf{u}_\gamma \cdot \tanh(\mathbf{W}_\gamma \mathbf{H}^t)) \quad (9)$$

The attentional vector for \mathbf{U}^t is finally computed as a weighted sum of its annotations, and, as shown in Fig. 4, is finally passed to a dense layer to obtain the utterance embedding $\mathbf{u}^t \in \mathbb{R}^{d_f}$:

$$\mathbf{u}^t = \text{dense} \left(\sum_{i=1}^{N+2} \gamma_i^t \mathbf{h}_i^t \right) \quad (10)$$

\mathbf{u}_γ can be interpreted as a learned representation of the “ideal word”, on average. The more similar a token vector is to this representation, the more attention the model pays to the token.

Context encoder: level 1. The pre and post-context vectors that we prepend and append to \mathbf{U}^t are obtained by aggregating information from the C utterances preceding and following \mathbf{U}^t :

$$\mathbf{u}_{\text{pre}}^t \leftarrow \text{aggregate}_{\text{pre}}(\{\mathbf{U}^{t-C}, \dots, \mathbf{U}^{t-1}\}) \quad (11)$$

$$\mathbf{u}_{\text{post}}^t \leftarrow \text{aggregate}_{\text{post}}(\{\mathbf{U}^{t+1}, \dots, \mathbf{U}^{t+C}\}) \quad (12)$$

where C , the context size, is a hyperparameter. Since $\mathbf{u}_{\text{pre}}^t$ and $\mathbf{u}_{\text{post}}^t$ will become part of utterance \mathbf{U}^t which is a sequence of token vectors, and fed to the RNN, we need them to live in the same space as any other token vector. This forbids the use of any nonlinear or dimension-changing transformation in aggregate, such as convolutional or recurrent operations. Therefore, we use self-attention only. More precisely, we propose a two-level hierarchical architecture that makes use of a different type of self-attention at each level (see left part of Fig. 4). The pre and post-context encoders share the exact

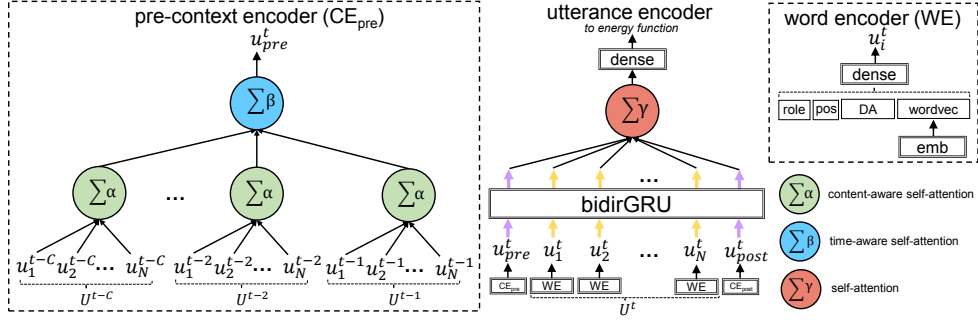


Figure 4: Our proposed utterance encoder G_W . Only the pre-context encoder is shown. C is the context size.

same architecture, so we only describe the pre-context encoder in what follows.

Content-aware self-attention. At level 1, we apply the same attention mechanism to each utterance in $\{\mathbf{U}^{t-C}, \dots, \mathbf{U}^{t-1}\}$. E.g., for \mathbf{U}^{t-1} :

$$\alpha^{t-1} = \text{softmax}\left(\mathbf{u}_\alpha \cdot \tanh\left(\mathbf{W}_\alpha \mathbf{U}^{t-1} + \mathbf{W}' \sum_{i=1}^N \mathbf{u}_i^t\right)\right) \quad (13)$$

This mechanism is the same as in Eq. 9, except for two differences. First, we operate directly on the matrix of token vectors of the previous utterance \mathbf{U}^{t-1} rather than on RNN annotations. Second, there is an extra input that consists of the element-wise sum of the token vectors of the current utterance \mathbf{U}^t . The latter modification is inspired by the coverage vectors used in translation and summarization to address under(over)-translation and repetition, e.g., (Tu et al., 2016; See et al., 2017). In our case, we hope that by letting the model know about the tokens in the current utterance \mathbf{U}^t , it will be able to extract complementary -rather than redundant- information from its context, and thus produce a richer embedding.

To recapitulate, the content-aware attention mechanism transforms the sequence of utterance matrices $\{\mathbf{U}^{t-C}, \dots, \mathbf{U}^{t-1}\} \in \mathbb{R}^{C \times N \times d}$ into a sequence of vectors $\{\mathbf{u}^{t-C}, \dots, \mathbf{u}^{t-1}\} \in \mathbb{R}^{C \times d}$. These vectors are then aggregated into a single pre-context vector $\mathbf{u}_{\text{pre}}^t \in \mathbb{R}^d$ as described next.

Note that since there is no inherent difference between preceding and following utterances³, we use the same content-aware self-attention mechanism for the pre and post contexts. This also gives us a more parsimonious and faster model. One should note, however, that the pre and post-context encoders still differ in terms of their time-aware attention mechanisms at level 2, described next.

³indeed, the latter become the former as we slide the window over the transcription

Context encoder: level 2. As can be seen in Fig. 2, two utterances close to each other in time are much more likely to be related (e.g., adjacency pair, elaboration...) than any two randomly selected utterances. To enable our model to capture such time dependence, we use the trainable universal time-decay attention mechanism of Su et al. (2018).

Time-aware self-attention. The mechanism combines three types of time-decay functions via weights w_i . The attentional coefficient for \mathbf{u}^{t-1} is:

$$\begin{aligned} \beta^{t-1} &= w_1 \beta^{\text{conv}^{t-1}} + w_2 \beta^{\text{lin}^{t-1}} + w_3 \beta^{\text{conc}^{t-1}} \quad (14) \\ &= \frac{w_1}{a(d^{t-1})^b} + w_2 [e d^{t-1} + k]^+ + \frac{w_3}{1 + \left(\frac{d^{t-1}}{D_0}\right)^l} \end{aligned}$$

where $[*]^+ = \max(*, 0)$ (ReLU), d^{t-1} is the offset between the positions of \mathbf{U}^{t-1} and \mathbf{U}^t , i.e., $d^{t-1} = |t - (t-1)| = 1$, and the w_i 's, a , b , e , k , D_0 , and l are scalar parameters learned during training.

The convex (conv), linear (lin), and concave (conc) terms respectively assume the strength of dependence to weaken rapidly, linearly, and slowly, as the distance in time increases. The post-context mechanism can be obtained by symmetry. It has different parameters.

5 Community detection

Once the utterance encoder G_W presented in Section 4 has been trained within the siamese or triplet meta-architecture presented in Section 3, it is used to project the summary-worthy utterances from a given test transcription to a compact embedding space. We assume that if training was successful, the distance in that space encodes community structure, so that a basic clustering algorithm such as k -means (MacQueen, 1967) is enough to capture it. However, since we need to detect overlapping communities, we use a probabilistic version of k -means, the Fuzzy c-Means (FCM) algorithm (Bezdek et al.,

1984). FCM returns a probability distribution over all communities for each utterance. More details are provided in App. E.

6 Experimental setup

6.1 Dataset

We experiment on the AMI corpus (McCowan et al., 2005), with the manual annotations v1.6.2. The corpus contains data for more than 100 meetings, in which participants play 4 roles within a design team whose task is to develop a prototype of TV remote control. Each meeting is associated with the annotations described in the introduction and shown in Fig. 2. There are 2368 unique abstractive communities in total, whose statistics are shown in Table 1. We adopt the officially suggested *scenario-only partition*⁴, which provides 97, 20, and 20 meetings respectively for training, validation and testing. We use manual transcriptions, and do not apply any particular preprocessing except filtering out specific ASR tags, such as `vocalsound`.

type	abstract	action	problem	decision	total
unique	1147	247	380	594	2368
disjoint	528	124	69	45	766
overlapping	349	17	163	149	678
singleton	49	162	38	244	493

Table 1: Statistics of abstractive communities.

6.2 Baselines

Full baseline details are provided in App. B.

- **Encoders.** First, we evaluate our utterance encoder against two encoders that are trained within the energy framework: (1) **LD** (Lee and Dernoncourt, 2016), a sequential sentence encoder developed for dialogue act classification; and (2) **HAN** (Yang et al., 2016), a hierarchical self-attentive network for document embedding. Note that to be fair, we ensure that both LD and HAN have access to context (see details in App. B).

We also compare our full pipeline against unsupervised and supervised systems.

- **Unsupervised systems.** In (1) **tf-idf**, we combine the TF-IDF vectors of the current utterance and the context utterances, each concatenated with their discourse features, and apply FCM. In (2) **w2v**, we repeat the same approach with the word2vec centroids of the words in each utterance. We also compare our full pipeline against **LCseg** (Galley et al., 2003), a lexical-cohesion based topic

⁴<http://groups.inf.ed.ac.uk/ami/corpus/datasets.shtml>

segmenter that directly clusters utterances without computing embeddings.

- **Supervised systems.** Finally, here, we use an approach similar to that of Murray et al. (2012). More precisely, we train a MLP to learn abstractive links between utterances, and then apply the CONGA community detection algorithm to the utterance graph.

We also considered 4 variants of our model: (1) **CA-S**: we replace the time-aware self-attention mechanism of the context encoder with basic self-attention. (2) **S-S**: we replace both the content-aware and the time-aware self-attention mechanisms of the context encoder with basic self-attention. (3) **(0,0)**: our model, without using the contextual encoder. (4) **(3,0)**: our model, using only pre-context, with a small window of 3, to enable fair comparison with the LD baseline.

6.3 Training details

Word encoder. Discourse features consist of two one-hot vectors of dimensions 4 and 16, respectively for speaker role and dialogue act. The positional feature is a scalar in $[0, 1]$, indicating the normalized position of the utterance in the transcription. We used the pre-trained vectors learned on the Google News corpus with word2vec by (Mikolov et al., 2013), and randomly initialized out-of-vocabulary words (1645 out of 12412). As a preprocessing step, we reduced the dimensionality of the pre-trained word vectors from 300 to 21 with PCA, in order to give equal importance to discourse and textual features. In the end, tokens are thus represented by a $d = 42$ -dimensional vector.

Layer sizes. For our model, and the LD and HAN baselines, we set $d_f = 32$ (output dimension of the final dense layer).

LD. We set $d1=3$ and $d2=0$, which is very close to (2,0), the best configuration reported in the original paper.

HAN. Again, for the sake of fairness, we give the HAN baseline access to contextual information, by feeding it the current utterance surrounded by the C_b preceding and C_b following utterances in the transcription, where C_b denotes the best context size reported in Section 7.

Training details. The exact same token representations and settings were used for our model, its variants, and the baselines. Models were trained on the training set for 30 epochs with the Adam (Kingma and Ba, 2015) optimizer. The best epoch

was selected as the one associated with the lowest validation loss. Batch size and dropout (Srivastava et al., 2014) were set to 16 and 0.5. Dropout was applied to the word embedding layer only. To account for randomness, we average results over 10 runs. Also, following (Hoffer and Ailon, 2015; Liu et al., 2019), we use a different, small subset of all possible triplets for training at each epoch (more precisely, 15594 triplets). This intelligently maximizes data usage while preventing overfitting. To enable fair comparison with the siamese approach, 15594 genuine and 15594 impostor pairs were sampled at the beginning of each epoch, since we consider that one triplet essentially equates one genuine pair and one impostor pair.

Performance evaluation. We evaluate performance at the distance and the clustering level, using respectively precision, recall, and F1 score at k , and the Omega Index (Collins and Dent, 1988) following the previous ACD work of Murray et al. (2012). For P, R, and F1, we evaluate the quality of the ranking of the closest utterances to a given query utterance. We use a fixed $k=10$ and also a variable k (denoted as $k=v$), where k is equal to the size of the community of the query utterance minus one. In that case, $P=R=F1$. More details and examples are given in Appendices C and D.

For the Omega Index, we report results with a fixed number of communities $|Q|=11$, and also a variable $|Q|$ ($|Q|=v$), where $|Q|$ is equal to the number of ground truth communities. More details and examples are given in App. C.

Due to the stochastic nature of the FCM algorithm, we select the run yielding the smallest objective function value over 20 runs.

7 Results

Context sizes. Larger contexts bring richer information, but increase the risk of considering unrelated utterances. Using our proposed encoder within the triplet meta-architecture, we tried different values of C on the validation set, under two settings: (pre, post) = $(C, 0)$, and (pre, post) = (C, C) . Results are shown in Fig. 5. We can observe that increasing C always brings improvement, with diminishing returns. Results also clearly show that considering the following utterances in addition to the preceding ones is useful. Note that the curves look similar for $F1@k = 10$. In the end, we selected (11,11) as our best context sizes.

Quantitative results. Final test set results are

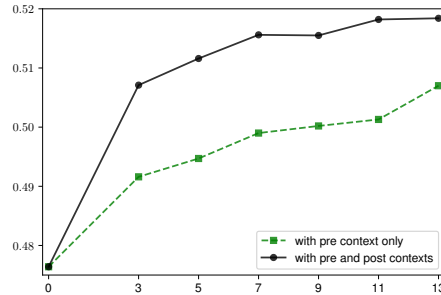


Figure 5: Impact of context size on the validation $P@k = v$, for our model trained within the triplet meta-architecture.

shown in Table 2. All variants of our model significantly outperform LD. While HAN is much stronger than LD, our model and its variants using best context sizes manage to outperform it everywhere, except in the siamese/ $P@k=v$ case (row j). One of the reasons for the superiority of our utterance encoder is probably that it considers contextual information *while encoding* the current utterance, while HAN and LD take as input the context utterances together with the current utterance, without distinguishing between them. Moreover, we use an attention mechanism dedicated to temporality, whereas HAN is only able to capture an implicit notion of time through the use of recurrence (RNN), and LD, with its dense layers, completely ignores it. Also, all variants of our model using best context sizes (11,11) outperform the ones using reduced (3,0) or no (0,0) context, regardless of the meta-architecture. This confirms the value added by our context encoder.

For siamese, our model outperforms its two variants (CA-S and S-S) for all metrics, indicating that both the content-aware and the time-aware self-attention mechanisms are useful. However, it is interesting to note that when training under the triplet configuration, the CA-S variant of our model is better, suggesting that in that case, the content-aware mechanism is beneficial, but the time-aware one is not.

LCseg (row m) and tf-idf (11,11) (row n3) are the best of all (un)supervised baseline systems, but both perform significantly worse than all energy-based approaches, highlighting that training with the energy framework is beneficial. In terms of Omega Index, supervised baseline systems are logically better than unsupervised ones.

w2v generally outperforms tf-idf when there is no context (rows k1,l1,n1,o1) or short context (k2,l2,n2,o2), but not with large contexts (k3,l3,n3,o3). Results also show that overall, using

				(pre, post)	P	P	R	F1	Omega index $\times 100$	
					@ $k = v$	@ $k = 10$			$ Q = v$	$ Q = 11$
Triplet	a1)	our model	(0, 0)	54.59	46.05	62.45	43.18	49.09	48.81	
	a2)	our model	(3, 0)	55.17	46.17	62.80	43.25	49.78	49.70	
	a3)	our model	(11, 11)	58.58	46.73	63.82	43.83	49.90	49.28	
	b)	our model (CA-S)	(11, 11)	59.52*	46.98*	64.01*	44.06*	50.11	49.73	
	c)	our model (S-S)	(11, 11)	58.96	46.81	63.65	43.87	49.59	49.88	
	d)	LD	(3, 0)	52.04	44.82	60.41	41.82	48.70	48.14	
e)	HAN	(11, 11)	58.72	45.76	62.60	42.89	49.32	48.88		
Siamese	f1)	our model	(0, 0)	53.01	45.10	60.97	42.12	50.56	49.65	
	f2)	our model	(3, 0)	53.78	45.54	61.33	42.48	51.01	50.00	
	f3)	our model	(11, 11)	56.64	46.47	62.54	43.40	52.44*	51.88*	
	g)	our model (CA-S)	(11, 11)	56.46	46.08	61.92	43.02	51.60	50.98	
	h)	our model (S-S)	(11, 11)	55.68	45.64	61.17	42.53	52.26	51.11	
	i)	LD	(3, 0)	52.13	44.83	60.85	41.86	51.18	50.70	
j)	HAN	(11, 11)	58.54	45.72	61.55	42.74	50.51	49.82		
Unsupervised	k1)	tf-idf	(0, 0)	29.28	26.67	34.69	24.19	13.12	13.66	
	k2)	tf-idf	(3, 0)	34.77	30.27	40.83	27.79	10.22	10.17	
	k3)	tf-idf	(11, 11)	58.94	43.94	61.36	41.45	38.09	39.47	
	l1)	w2v	(0, 0)	29.02	27.46	37.39	25.11	13.89	13.50	
	l2)	w2v	(3, 0)	34.11	29.92	39.55	27.32	10.61	10.77	
	l3)	w2v	(11, 11)	58.30	44.08	61.59	41.59	37.75	38.28	
	m)	LCSeg	-	-	-	-	-	38.98	41.57	
Supervised	n1)	tf-idf	(0, 0)	-	-	-	-	25.04	25.14	
	n2)	tf-idf	(3, 0)	-	-	-	-	27.33	26.95	
	n3)	tf-idf	(11, 11)	-	-	-	-	45.26	44.91	
	o1)	w2v	(0, 0)	-	-	-	-	25.32	25.25	
	o2)	w2v	(3, 0)	-	-	-	-	29.14	29.02	
	o3)	w2v	(11, 11)	-	-	-	-	43.31	43.08	

Table 2: Results (averaged over 10 runs). *: best score per column. **Bold**: best score per section. -: does not apply as the method does not produce utterance embeddings.

larger contexts always brings improvement.

Qualitative results. We visualize in App. A that the three self-attention mechanisms behave in a cooperative manner to produce a meaningful utterance representation. We also inspect the closest utterances to a given query utterance in App. D. We also visualize the time-aware self-attention coefficients in Fig. 6, and find that interestingly, the distributions over the pre and post-context are not symmetric. Indeed, only the utterances immediately following U^t ($t + 1 \rightarrow t + 5$) seem to matter, while the attention weights are much more uniform across the utterances preceding U^t . This suggests that in dialogues, considering a long history of preceding utterances helps understanding the current one. Overall, the three terms (see Eq. 14) altogether do produce a `universal` function that decreases as time distance increases, which is in accordance with intuition.

Simplified task. Finally, we also experimented on a much simpler task, where only the communities of type ABSTRACT were considered. This makes ACD much simpler, because most of the overlapping communities are of the other types (see Table 1). For this simplified task, we have 1147 unique

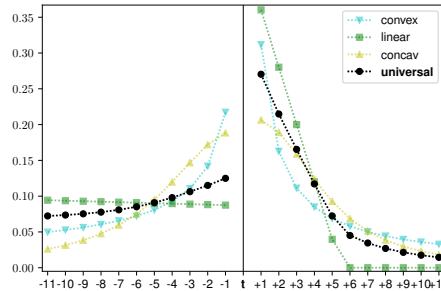


Figure 6: Normalized time-aware self-attention weights for pre and post-contexts, averaged over 10 runs.

communities, of which 78.99% are disjoint. our model achieves 72.09 in terms of $P@k = v$ and 55.67 in terms of Omega Index when $|Q| = v$. $P, R, F1@k = 15$ are respectively equal to 55.07, 74.37, and 54.00, and the Omega Index is 54.30 when $|Q| = 8$.

8 Conclusion

This paper proposes one of the first applications of energy-based learning to ACD. Using the siamese and triplet meta-architectures, we showed that our novel contextual utterance encoder learns better distance and communities than state-of-the-art competitors.

Acknowledgments

This research was supported in part by the OpenPaaS::NG and LinTo (Lorré et al., 2019) projects.

References

- Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. [Generating abstractive summaries from meeting transcripts](#). In *Proceedings of the 2015 ACM Symposium on Document Engineering, DocEng '15*, pages 51–60, New York, NY, USA.
- James C. Bezdek, Robert Ehrlich, and William Full. 1984. [Fcm: The fuzzy c-means clustering algorithm](#). *Computers & Geosciences*, 10(2):191 – 203.
- Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. [Signature verification using a "siamese" time delay neural network](#). In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 737–744. Morgan-Kaufmann.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. [Learning phrase representations using RNN encoder–decoder for statistical machine translation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. ACL.
- S. Chopra, R. Hadsell, and Y. 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'05)*, volume 1, pages 539–546 vol. 1.
- Linda M. Collins and Clyde W. Dent. 1988. [Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions](#). *Multivariate Behavioral Research*, 23(2):231–242. PMID: 26764947.
- Michel Galley, Kathleen R. McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. [Discourse segmentation of multi-party conversation](#). In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 562–569, Sapporo, Japan. ACL.
- Nikhil Garg, Benoît Favre, Korbinian Riedhammer, and Dilek Hakkani-Tür. 2009. [Clusterrank: a graph based method for meeting summarization](#). In *INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association, Brighton, United Kingdom, September 6-10, 2009*, pages 1499–1502. ISCA.
- Steve Gregory. 2007. [An algorithm to find overlapping community structure in networks](#). In *Knowledge Discovery in Databases: PKDD 2007*, pages 91–102, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Elad Hoffer and Nir Ailon. 2015. [Deep metric learning using triplet network](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Yann Lecun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. 2006. [A tutorial on energy-based learning](#). In *Predicting structured data*. MIT Press.
- Yann LeCun and Fu Jie Huang. 2005. [Loss functions for discriminative training of energy-based models](#). In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, AISTATS 2005, Bridgetown, Barbados, January 6-8, 2005*. Society for Artificial Intelligence and Statistics.
- Ji Young Lee and Franck Dernoncourt. 2016. [Sequential short-text classification with recurrent and convolutional neural networks](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–520. ACL.
- Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. [A structured self-attentive sentence embedding](#). In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.
- Jinchao Liu, Stuart J. Gibson, James Mills, and Margarita Osadchy. 2019. [Dynamic spectrum matching with one-shot learning](#). *Chemometrics and Intelligent Laboratory Systems*, 184:175 – 181.
- Jean-Pierre Lorré, Isabelle Ferrané, Francisco Madrigal, Michalis Vazirgiannis, and Christophe Bourguinat. 2019. [Linto: Assistant vocal open-source respectueux des données personnelles pour les réunions d'entreprise](#). *APIA*, page 63.
- J. MacQueen. 1967. [Some methods for classification and analysis of multivariate observations](#). In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif. University of California Press.
- Iain McCowan, Jean Carletta, W Kraaij, S Ashby, S Bourban, M Flynn, M Guillemot, T Hain, J Kadlec, V Karaiskos, et al. 2005. [The ami meeting corpus](#). In *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, volume 88.

- Yashar Mehdad, Giuseppe Carenini, Frank Tompa, and Raymond T. NG. 2013. [Abstractive meeting summarization with entailment and fusion](#). In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 136–146. ACL.
- Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. [Exploiting similarities among languages for machine translation](#). *CoRR*, abs/1309.4168.
- Jonas Mueller and Aditya Thyagarajan. 2016. [Siamese recurrent architectures for learning sentence similarity](#). In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2786–2792. AAAI Press.
- Gabriel Murray, Giuseppe Carenini, and Raymond Ng. 2012. [Using the omega index for evaluating abstractive community detection](#). In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 10–18, Montréal, Canada. ACL.
- Gabriel Murray, Steve Renals, and Jean Carletta. 2005. [Extractive summarization of meeting recordings](#). In *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*, pages 593–596. ISCA.
- Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. [Learning text similarity with siamese recurrent networks](#). In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 148–157. ACL.
- Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. 2014. [A template-based abstractive meeting summarization: Leveraging summary and source text relationships](#). In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 45–53. ACL.
- F. Schroff, D. Kalenichenko, and J. Philbin. 2015. [Facenet: A unified embedding for face recognition and clustering](#). In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 815–823.
- Veit Schwämmle and Ole Nørregaard Jensen. 2010. [A simple and fast method to determine the parameters for fuzzy c-means cluster analysis](#). *Bioinformatics*, 26(22):2841–2848.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. ACL.
- Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. [Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 664–674. ACL.
- Karan Singla, Evgeny Stepanov, Ali Orkan Bayer, Giuseppe Carenini, and Giuseppe Riccardi. 2017. [Automatic community creation for abstractive spoken conversations summarization](#). In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 43–47. ACL.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. [Dropout: A simple way to prevent neural networks from overfitting](#). *Journal of Machine Learning Research*, 15:1929–1958.
- Shang-Yu Su, Pei-Chieh Yuan, and Yun-Nung Chen. 2018. [How time matters: Learning time-decay attention for contextual spoken language understanding in dialogues](#). 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 2133–2142, New Orleans, Louisiana. ACL.
- Antoine Tixier, Polykarpos Meladianos, and Michalis Vazirgiannis. 2017. [Combining graph degeneracy and submodularity for unsupervised extractive summarization](#). In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 48–58. ACL.
- Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. [Modeling coverage for neural machine translation](#). *arXiv preprint arXiv:1601.04811*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.
- Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. [Learning fine-grained image similarity with deep ranking](#). In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14*, pages 1386–1393, Washington, DC, USA. IEEE Computer Society.
- Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. [Exploiting cross-sentence context for neural machine translation](#). *arXiv preprint arXiv:1704.04347*.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. [Hierarchical attention networks for document classification](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489. ACL.

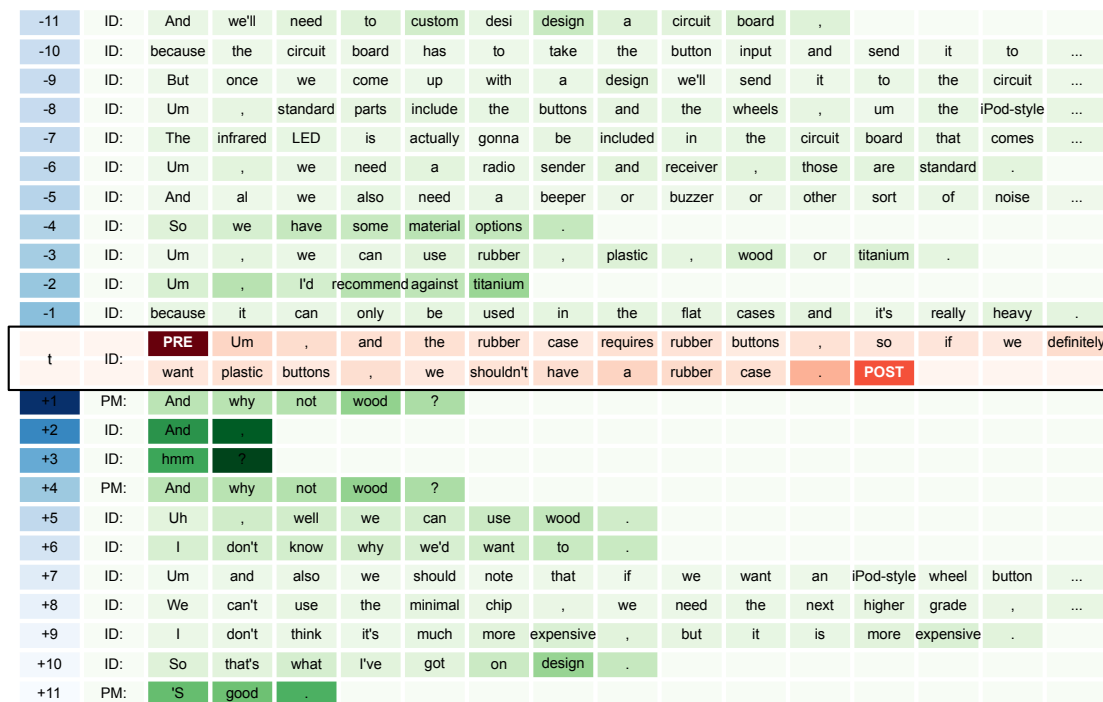


Figure 7: Visualization of attention distributions around an utterance from the ES2011c meeting. Some utterances are truncated for readability.

Appendices

A Attention visualization

The aim of this section is to show, with an example, what the three self-attention mechanisms pay attention to while encoding the current utterance U^t (here, an utterance from the ES2011c *validation* meeting). Fig. 7 shows the attention distributions over U^t (highlighted by the black frame), and over its pre-context $\{U^{t-1}, \dots, U^{t-11}\}$ and post-context $\{U^{t+1}, \dots, U^{t+11}\}$ utterances. We use three colors that are consistent with the ones used in Fig. 4 to denote the three different attention mechanisms: green for content-aware (α), blue for time-aware (β), and red for basic self-attention (γ). Remember that α and β are both in the context encoder, while γ is in the utterance encoder. Color shades indicate attention intensity (the darker, the stronger).

We can observe in Fig. 7 that:

- The content-aware self-attention mechanism α (green) focuses on the informative and complementary words in the contexts that are central to understanding the utterance at time t , such as: “custom”, “design” from U^{t-11} , “material” from U^{t-4} , “recommend”, “titanium” from U^{t-2} , “wood” from U^{t+1} , etc.

- The time-aware self-attention mechanism β (blue) places more importance over the context utterances that are close to U^t , i.e., the importance decreases when the time distance increases. However, the patterns are different for the pre and post-contexts (see Fig. 6).

- The self-attention mechanism γ (red) focuses mainly on the special pre-context token PRE, meaning that the pre-context is more important than the post-context in the example considered. Generally speaking, the pre and post-context tokens contain richer information than any token from the current utterance, as the context tokens originate from the fusion of $\{U^{t-11}, \dots, U^t, \dots, U^{t+11}\}$. It is thus possible that the utterance encoder has learned to always pay more attention to these information-rich tokens than to any regular token.

- It is also interesting to note that considerable attention is being paid to punctuation marks. This makes sense, since they are important pieces of information indicative of utterance type (e.g., statement or question).

To summarize, the visualization results show that the three self-attention mechanisms of our model are able to adaptively focus on different information, in order to cooperatively produce a meaningful representation.

B Baselines

B.1 Baseline encoders

- **LD** (Lee and Deroncourt, 2016) is a sequential sentence encoder developed for dialogue act classification. The model takes into account a fixed number of utterances from the pre-context when classifying the current one. More precisely, CNN or RNN with max-pooling is first applied separately to the current utterance and each pre-context utterance, and the resulting vectors are then aggregated through two levels of dense layers, based on two hyper-parameters, $d1$ and $d2$, which represent the history size at level 1 and level 2 (respectively). Although the original paper reported that the CNN encoder slightly outperforms the RNN one (for DA classification), in our experiments, we used the RNN variant, since our model and the HAN baseline are RNN-based. Note that here, we used LSTM cells as Lee and Deroncourt (2016) reported them to work better than GRU cells in their experiments.

- **HAN** (Yang et al., 2016). The Hierarchical Attention Network, developed for document classification, is a two-level architecture, where at level 1, each sentence in the document is separately encoded by the same sentence encoder, resulting in a sequence of sentence vectors. That sequence is then processed at level 2 by the document encoder which returns a single vector representing the entire document. The sentence and document encoders are both self-attentional bidirectional Recurrent Neural Networks (RNNs), with different parameters. We give HAN access to contextual information by feeding it the current utterance surrounded by the C_b preceding and C_b following utterances in the transcription, where C_b denotes the best context size reported in Section 7.

B.2 Unsupervised baseline systems

- **tf-idf**. A TF-IDF vector is used as the utterance embedding, compressed to a dimension of 21 with PCA, and concatenated with the 21-dimensional discourse feature vector, thus forming a vector of dimension $d = 42$. This vector is then again compressed to a $d_f = 32$ -dimensional vector. The compression steps are applied for consistency with the energy-based systems, in which textual and discourse features have the same dimensionality $d/2 = 21$, and the output of the utterance encoder is d_f -dimensional (see Subsection 6.3). To make

this baseline context-aware, the embeddings of the current utterance and the context utterances are averaged. In the end, FCM is applied. Note that the TF-IDF vocabulary is obtained from the entire conversation, giving this baseline a competitive advantage over the others, which never have access to the full transcription.

- **w2v**. Identical to the previous baseline, but using the average of the word2vec vectors of a given utterance instead of TF-IDF vector.

- **LCseg** is an unsupervised system adapted from previous work (Oya et al., 2014; Banerjee et al., 2015; Singla et al., 2017), in which disjoint topic segments are assumed to be abstractive communities. A lexical-cohesion based topic segmenter LCseg (Galley et al., 2003) is first applied on transcriptions to get the desired number of segments ($|Q| = v/11$), and then only summary-worthy utterances within segments are retained for evaluation.

B.3 Supervised baseline systems

As discussed in the literature review (see Section 2), original approaches to ACD (Murray et al., 2012; Mehdad et al., 2013) are supervised and non energy-based. They have no publicly available implementations, and are hard to precisely reimplement due to lack of details about handcrafted features and dependency on external textual entailment corpora. Nevertheless, we implemented two baselines similar in spirit, taking as input the representations produced by the tf-idf and w2v unsupervised baselines previously described. More precisely, the two d_f -dimensional representations of a pair of utterances are fed into a 3-layer feed-forward neural network (with $2d_f$, d_f , and 1 hidden units) which is trained on the task of predicting whether the two utterances belong to the same abstractive community or not (binary classification task). Then, like in the aforelisted studies, an utterance graph is built, where utterances are linked based on the predictions of the MLP. Finally, the CONGA algorithm (Gregory, 2007), an extension of the well-known Girvan-Newman algorithm, is applied to detect overlapping communities in the utterance graph.

C Performance evaluation

We evaluate performance at the distance and the clustering level.

C.1 Distance

First, we test whether the distance in the final embedding space is meaningful. To do so, for a given *query* utterance, we rank all other utterances in decreasing order of similarity with the query. We then use precision, recall, and F1 score at k to evaluate the quality of the ranking. A detailed example was provided in App. D.

Singleton communities are excluded from the evaluation at this stage. We set $k=10$, which is equal to the average number of non-singleton communities minus one (since the query utterance cannot be part of the results). We also report results for a variable k ($k=v$), where k is equal to the size of the community of the query utterance minus one. In that case, $P=R=F1$.

The same procedure is repeated for all utterances. To account for differences in community size, scores are first averaged at the community-level, and then at the meeting-level. Note that the distance is Euclidean for triplet and Manhattan for siamese (see Subsections 3.2 and 3.3).

C.2 Clustering

Second, we compare our community assignments to the human ground truth using the Omega Index (Collins and Dent, 1988), a standard metric for comparing non-disjoint clustering, used in the ACD literature (Murray et al., 2012). The Omega Index evaluates the degree of *agreement* between two clustering solutions based on *pairs* of objects being clustered. Two solutions s_1 and s_2 are considered to agree on a given pair of objects, if two objects are placed by both solutions in *exactly* the same *number of communities* (possibly zero). The Omega Index ω is computed as shown in Eq. 15. The numerator is the observed agreement ω_{obs} adjusted by expected (chance) agreement ω_{exp} , while the denominator is the perfect agreement (value equals to 1) adjusted by expected agreement.

$$\omega(s_1, s_2) = \frac{\omega_{obs}(s_1, s_2) - \omega_{exp}(s_1, s_2)}{1 - \omega_{exp}(s_1, s_2)} \quad (15)$$

Observed and expected agreements are calculated as below:

$$\omega_{obs}(s_1, s_2) = \frac{1}{N_{total}} \sum_{j=0}^{\min(J,K)} A_j \quad (16)$$

$$\omega_{exp}(s_1, s_2) = \frac{1}{N_{total}^2} \sum_{j=0}^{\min(J,K)} N_{j1}N_{j2} \quad (17)$$

where A_j is the number of pairs agreed to be assigned to j number of communities by both solutions, N_{j1} is the number of pairs assigned to j communities in s_1 , N_{j2} is the number of pairs assigned to j communities in s_2 , J and K represent respectively the maximum number of communities in which any pair of objects appear together in solutions s_1 and s_2 , and $N_{total} = n(n-1)/2$ is the total number of pairs constructed over n number of objects. To give an example, consider two clustering solutions for 5 objects:

$$s_1 = \{\{a, b, c\}, \{b, c, d\}, \{c, d, e\}, \{c, d\}\}$$

$$s_2 = \{\{a, b, c, d\}, \{b, c, d, e\}\}$$

Solutions are transformed into the table 3, from what we can obtain $N_{total} = 10$, $J = 3$, $K = 2$, $\min(J, K) = 2$. Two solutions agree to place (a, e) together in no community, the pairs (a, b) , (a, c) , (c, e) and (d, e) in one community, and the pair (b, c) in two communities. We have $A_0 = 1$, $A_1 = 4$, $A_2 = 1$. Thus the observed agreement is $(1 + 4 + 1)/10 = 0.6$. Since $N_{01} = 3$, $N_{11} = 5$, $N_{21} = 1$ and $N_{02} = 1$, $N_{12} = 6$, $N_{22} = 3$, the expected agreement then is $(3 * 1 + 5 * 6 + 1 * 3)/10^2 = 0.36$. Finally, Omega Index for this simple example is computed as: $\omega(s_1, s_2) = (0.6 - 0.36)/(1 - 0.36) = 0.375$.

	solution s_1	solution s_2	solutions s_1 and s_2
	#communities the pair is assigned	#communities the pair is assigned	agree on the pair?
(a, b)	1	1	yes
(a, c)	1	1	yes
(a, d)	0	1	no
(a, e)	0	0	yes
(b, c)	2	2	yes
(b, d)	1	2	no
(b, e)	0	1	no
(c, d)	3	2	no
(c, e)	1	1	yes
(d, e)	1	1	yes

Table 3: Omega Index example.

Since FCM yields a probability distribution over communities for each utterance, we need to use a threshold to assign a given utterance to one or more communities. We selected 0.2 after trying multiple values in $[0, 0.5]$ with steps of 0.05 on the validation set. Whenever one or more utterances were not assigned to any community, we merged them into a new community. Furthermore, we set the number of clusters $|Q|$ to 11, which corresponds to the average number of ground truth communities per meeting (after merging). We also report results with a variable $|Q|$ equal to the number of ground truth communities (variant denoted by $|Q| = v$).

dist	pos	DA	role	text
0	<i>t</i>	inf	ID	Um , and the rubber case requires rubber buttons , so if we definitely want plastic buttons , we shouldn't have a rubber case .
0.11	-3	inf	ID	Um , we can use rubber , plastic , wood or titanium .
0.12	-5	inf	ID	And al we also need a beeper or buzzer or other sort of noise thing for locating the remote .
0.38	-2	sug	ID	Um , I'd recommend against titanium
0.42	+7	inf	ID	Um and also we should note that if we want an iPod-style wheel button , it's gonna require a m qu slightly more expensive chip .
0.54	+5	ass	ID	Uh , well we can use wood .
0.57	-8	inf	ID	Um , standard parts include the buttons and the wheels , um the iPod-style wheel .
0.68	+6	ass	ID	I don't know why we'd want to .
0.96	-11	inf	ID	And we'll need to custom desi design a circuit board ,
1.26	-13	inf	ID	Um , I assume we'll be custom designing our case ,
1.27	-14	inf	ID	Um , so we need some custom design parts , and other parts we'll just use standard .
1.43	-17	inf	ID	So I've been looking at the components design .
1.66	+12	off	ME	Um , can I do next ? Because I have to say something about the material
2.24	+18	inf	ME	and the findings are that the first thing to aim for is a fashion uh , fancy look and feel .
2.57	+19	inf	ME	Um . Next comes technologic technology and the innovations to do with that .
3.21	+20	inf	ME	And th last thing is the easy to use um factor .
3.92	+69	inf	UI	Uh , so people are going to be looking at this little screen .
4.02	+92	inf	ME	But the screen can come up on the telly , the she said .
...				
8.81	+623	inf	ID	It didn't give me any actual cost .
8.84	+622	inf	ID	All it said was it gave sort of relative , some chips are more expensive than others , sort of things .
8.89	+616	inf	ME	So if you throw it , it's gonna store loads of energy , and you don't need to buy a battery because they're quite f I find them annoying .
9.00	+617	sug	ME	But we need to find cost .
9.06	+621	el.inf	ME	Does anyone have costs on the on the web ?
9.95	+652	inf	PM	And you're gonna be doing protu product evaluation .
9.96	+650	inf	PM	Oh when we move on , you two are going to be playing with play-dough .
10.15	+651	inf	PM	Um , and working on the look and feel of the design and user interface design .

Table 4: Ranking example.

Note that since FCM does not return nested groupings, we merged the ground truth communities nested under the same community.

D Ranking example

For the same utterance from the ES2011c meeting as used in App. A, we show below in Table 4 the closest and furthest utterances, in terms of Euclidean distance in the embedding space. Recall that meeting ES2011c belongs to the *validation* set. Utterances belonging to the ground truth community of the query utterance are shown in bold. Roles are ID: industrial designer, ME: marketing expert, UI: user interface designer, PM: project manager. For this example, $P@k = v$ is equal to 77.78 (where $v = 9$), and P , R , and $F1@k$ are 80.00, 88.89, 84.21 respectively (where $k = 10$).

We can see that semantic similarity obviously

plays a role, as most of the closest utterances are about buttons and materials. But other parameters come into play. E.g., the utterances **And al we also need a beeper or buzzer or other sort of noise thing for locating the remote**, **and I don't know why we'd want to**, respectively ranked 2nd and 7th, are not semantically related to the query utterance. Such utterances might be placed close to the query utterance based on their positional and discourse features (speaker role and dialogue act), but also because their contexts are similar. The community where the query utterance belongs to (utterances shown in bold in the table above) is associated with the following sentence in the human abstractive summary: The Industrial Designer gave her presentation on components and discussed which would have to be custom-made and which were standard.

E FCM algorithm

The goal of the Fuzzy c-Means algorithm or FCM (Bezdek et al., 1984) is to minimize the weighted within group sum of squared error objective function:

$$J(M, Q) = \sum_{q=1}^{|Q|} \sum_{t=1}^T (m_{qt})^{fuz} \|\mathbf{u}^t - \mathbf{c}_q\|_2^2 \quad (18)$$

where M and Q are the sets of membership probability distributions and community centroid vectors, $m_{qt} \in [0, 1]$ is the probability that the t -th utterance belongs to the q -th community (with $\sum_{q=1}^{|Q|} m_{qt} = 1$), fuz is a parameter that controls the amount of fuzziness, $\|\cdot\|_2$ denotes the Euclidean distance in the triplet case (we replace it with Manhattan distance $\|\cdot\|_1$ in the siamese case), \mathbf{u}^t is the t -th utterance vector, and \mathbf{c}_q is the q -th community centroid vector.

M and Q are iteratively updated with equations:

$$m_{qt} = \left(\sum_{j=1}^{|Q|} \left(\frac{\|\mathbf{u}^t - \mathbf{c}_q\|_2}{\|\mathbf{u}^t - \mathbf{c}_j\|_2} \right)^{\frac{2}{fuz-1}} \right)^{-1} \quad (19)$$

$$\mathbf{c}_q = \frac{\sum_{t=1}^T (m_{qt})^{fuz} \mathbf{u}^t}{\sum_{t=1}^T (m_{qt})^{fuz}} \quad (20)$$

When $fuz \rightarrow +\infty$, $\forall q \in |Q|$, $\forall t \in T$, m_{qt} tends to be equal to $1/|Q|$, thus utterances have identical membership to each community. While when $fuz \rightarrow 1$, FCM becomes equivalent to traditional k -means, in which m_{qt} is either 0 or 1 for a given utterance \mathbf{u}^t and community centroid \mathbf{c}_q . Usually in practice, $fuz = 2$ (Schwämmle and Jensen, 2010). Learning stops when the maximum number of iterations is reached or $J(M, Q)$ decreases by less than a predefined threshold.

F On our choice of loss functionals

The softmax triplet loss (STL) performed better in our experiments than the margin-based loss used in (Schroff et al., 2015) and (Wang et al., 2014). One of the reasons may be that STL is able to capture a finer notion of distance. Indeed, with a margin-based loss, the Euclidean distance between the anchor and the negative (let us compactly denote it as d^-) need to satisfy $d^- > d^+ + m$, where m is the margin (see Fig. 8a). In other words, the distance between the positive and the negative is at least m , when all three points are aligned.

However, the objective of STL is simply $d^- > d^+$, without imposing an absolute lower bound on the

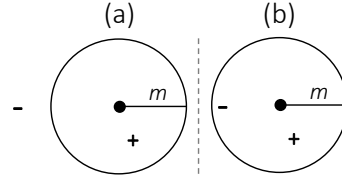


Figure 8: •, -, + denote anchor, negative, and positive.

distance between positives and negatives. That is, only the distance ratio is of interest (see Fig. 8b), which gives more freedom to the model.

For consistency, we also adopt a margin-free loss functional for siamese (MSE, see Eq. 5). It performed better than the traditional contrastive loss (Chopra et al., 2005; Neculoiu et al., 2016) in early experiments.