Spoken Language Understanding for Abstractive Meeting Summarization

Guokan Shang

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAX011

Thèse de doctorat

Spoken Language Understanding for Abstractive Meeting Summarization

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École Polytechnique

École doctorale n°626 : l'École Doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Informatique, Données et Intelligence Artificielle

Thèse présentée et soutenue à Palaiseau, le 28/01/2021, par

Guokan Shang

Composition du Jury :

François Jacquenet Professor, Université Jean Monnet Saint-Étienne	Président
Xuedong Huang Technical Fellow & Chief Technology Officer, Microsoft	Rapporteur
Xiaojun Wan Professor, Peking University	Rapporteur
Chloé Clavel Professor, Télécom Paris	Examinateur
Florian Boudin Associate Professor, Université de Nantes	Examinateur
Marianna Apidianaki Senior Researcher, University of Helsinki	Examinateur
Michalis Vazirgiannis Professor, École Polytechnique	Directeur de thèse
Jean-Pierre Lorré R&D Director, LINAGORA	Co-encadrant de thèse

ABSTRACT

With the impressive progress that has been made in transcribing spoken language, it is becoming increasingly possible to exploit transcribed data for tasks that require comprehension of what is said in a conversation. The work in this dissertation, carried out in the context of a project devoted to the development of a meeting assistant, contributes to ongoing efforts to teach machines to understand multi-party meeting speech. In particular, we have focused on the challenge of automatically generating abstractive meeting summaries, which would be of great value to individuals as well as organizations. Our research has been conducted to address three specific tasks, each of which is addressed in a separate chapter of this dissertation.

We first present our results on Abstractive Meeting Summarization (AMS), which aims to take a meeting transcription as input and produce an abstractive summary as output. We introduce a fully unsupervised framework for this task based on multi-sentence compression and budgeted submodular maximization. We also leverage recent advances in word embeddings and graph degeneracy applied to NLP, to take exterior semantic knowledge into account and to design custom diversity and informativeness measures. Experiments show that our system improves on the state-of-the-art, and generates reasonably grammatical abstractive summaries despite taking noisy utterances as input and not relying on any annotations.

Next, we discuss our work on Dialogue Act Classification (DAC), whose goal is to assign each utterance in a discourse a label that represents its communicative intention. DAC yields annotations that are useful for a wide variety of tasks, including AMS. We propose a modified neural Conditional Random Field (CRF) layer that takes into account not only the sequence of utterances in a discourse, but also speaker information and in particular, whether there has been a change of speaker from one utterance to the next. Experiments and visualizations show that our modified CRF layer outperforms the original one on DAC and learns meaningful transition patterns between dialogue acts conditioned on speaker-change.

The third part of the dissertation focuses on Abstractive Community Detection (ACD), a sub-task of AMS, in which utterances in a conversation are grouped according to whether they can be jointly summarized by a common abstractive sentence. We provide a novel approach to ACD in which we first introduce a neural contextual utterance encoder featuring three types of self-attention mechanisms and then train it using the siamese and triplet energy-based meta-architectures. We further propose a general sampling scheme that enables the triplet ar-

chitecture to capture subtle patterns (e.g., overlapping and nested clusters). Experiments and visualizations show that our system improves on the state-of-the-art and that our triplet sampling scheme is effective.

RÉSUMÉ

Grâce aux progrès impressionnants qui ont été réalisés dans la transcription du langage parlé, il est de plus en plus possible d'exploiter les données transcrites pour des tâches qui requièrent la compréhension de ce que l'on dit dans une conversation. Le travail présenté dans cette thèse, réalisé dans le cadre d'un projet consacré au développement d'un assistant de réunion, contribue aux efforts en cours pour apprendre aux machines à comprendre les dialogues des réunions multipartites. En particulier, nous nous sommes concentrés sur le défi de générer automatiquement les résumés abstractifs de réunion, ce qui serait d'une grande valeur pour les individus comme pour les organisations. Notre recherche a été menée pour aborder trois tâches spécifiques, dont chacune est abordée dans un chapitre séparé de cette thèse.

Nous présentons tout d'abord nos résultats sur le Résumé Abstractif de Réunion (RAR), qui consiste à prendre une transcription de réunion comme entrée et à produire un résumé abstractif comme sortie. Nous introduisons une approche entièrement non-supervisée pour cette tâche, basée sur la compression multi-phrases et la maximisation sous-modulaire budgétisée. Nous tirons également parti des progrès récents en vecteurs de mots et dégénérescence de graphes appliqués au TAL, afin de prendre en compte les connaissances sémantiques extérieures et de concevoir de nouvelles mesures de diversité et d'informativité. Les expérimentations montrent que notre système améliore l'état de l'art et génère des résumés abstractifs raisonnablement grammaticaux, même s'il prend en entrée des énoncés bruyants et ne s'appuie sur aucune annotation.

Ensuite, nous discutons de notre travail sur la Classification en Actes de Dialogue (CAD), dont le but est d'attribuer à chaque énoncé d'un discours une étiquette qui représente son intention communicative. La CAD produit des annotations qui sont utiles pour une grande variété de tâches, y compris le RAR. Nous proposons une couche neuronale modifiée de Champ Aléatoire Conditionnel (CAC) qui prend en compte non seulement la séquence des énoncés dans un discours, mais aussi les informations sur les locuteurs et en particulier, s'il y a eu un changement de locuteur d'un énoncé à l'autre. Les expérimentations et les visualisations montrent que notre couche CAC modifiée est plus performante que la couche originale sur la CAD et apprend des schémas de transition faisant sens entre les actes de dialogue conditionnés par le changement de locuteur.

La troisième partie de la thèse porte sur la Détection de Communauté Abstractive (DCA), une sous-tâche du RAR, dans laquelle les

énoncés d'une conversation sont regroupés selon qu'ils peuvent être résumés conjointement par une phrase abstractive commune. Nous proposons une nouvelle approche de la DCA dans laquelle nous introduisons d'abord un encodeur neuronal contextuel d'énoncé qui comporte trois types de mécanismes d'auto-attention, puis nous l'entraînons en utilisant les méta-architectures siamoise et triplette basées sur l'énergie. Nous proposons en outre une méthode d'échantillonnage générale qui permet à l'architecture triplette de capturer des motifs subtils (par exemple, des groupes qui se chevauchent et s'emboîtent). Les expérimentations et les visualisations montrent que notre système améliore l'état de l'art et que notre méthode d'échantillonnage des triplets est efficace.

PUBLICATIONS

The following publications are included in parts or in an extended version in this thesis:

Shang, Guokan, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré (July 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)*. Melbourne, Australia: Association for Computational Linguistics, pp. 664–674. DOI: 10.18653/v1/P18-1062. URL: https://www.aclweb.org/anthology/P18-1062.

Shang, Guokan, Antoine Tixier, Michalis Vazirgiannis, and Jean-Pierre Lorré (Dec. 2020a). « Energy-based Self-attentive Learning of Abstractive Communities for Spoken Language Understanding. » In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 313–327. URL: https://www.aclweb.org/anthology/2020.aacl-main.34.

Shang, Guokan, Antoine Tixier, Michalis Vazirgiannis, and Jean-Pierre Lorré (Dec. 2020b). « Speaker-change Aware CRF for Dialogue Act Classification. » In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 450–464. URL: https://www.aclweb.org/anthology/2020.coling-main.40.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my supervisors, Prof. **Michalis Vazirgiannis** and Mr. **Jean-Pierre Lorré**, without whom this dissertation would not have been possible. They gave me the precious opportunity to have my Ph.D. conducted in both academic and industrial contexts: at the École Polytechnique and at LINAGORA, which has allowed me to gain valuable experience in both worlds and to understand their different research aims and approaches. They are excellent advisors: on a professional level, our various contributions and my involvement in the research community were born out of our numerous discussions, and on a personal level, they have treated me as a friend, made me better in all aspects, and supported and encouraged me to achieve this goal. I also gratefully acknowledge the support of the French CIFRE Ph.D. fellowship that funded my work.

Secondly, I would like to thank my close collaborator Dr. **Antoine Tixier** who worked with me and co-authored all papers. From the bottom of my heart, I know how fortunate I have been to have had him by my side, fighting all the scientific problems that we have encountered in our research. I'm always amazed by, and appreciative of, his academic rigor and pursuit, from which I have learned so much. I would also like to thank, in particular, Dr. **Julie Hunter**, for her great and continuous efforts made in helping me prepare and review my dissertation, as well as for her insights provided to me in the area of linguistics through our many lively discussions.

Furthermore, I want to express my gratitude to the distinguished researchers: Dr. **Xuedong Huang**, Prof. **Xiaojun Wan**, Prof. **François Jacquenet**, Prof. **Chloé Clavel**, Prof. **Florian Boudin**, and Dr. **Marianna Apidianaki**, for agreeing to be part of the jury of my Ph.D. defense, for their valuable feedback on my work and for their thought-provoking questions during my defense. A special thanks goes to Dr. Xuedong Huang and Prof. Xiaojun Wan who reviewed this dissertation and gave detailed insightful comments about my research.

Additionally, I would like to thank my wonderful colleagues that I have been truly lucky to interact with, current and past, for all I learned from them, and especially,
from **École Polytechnique**: George Dasoulas, Moussa Eddine Kamal, Sammy Khalife, Panagiotis Korvesis, Stratis Limnios, Johannes Lutzeyer, Fragkiskos Malliaros, Giannis Nikolentzos, Olivier Pallanca, George Panagopoulos, Yang Qiu, Jesse Read, Maria Rossi, Guillaume Salha, Konstantinos Skianis, Nikolaos Tziortziotis, Changmin Wu, Christos Xypolopoulos, and Chenhui Zhang,

from **LINAGORA**: Sonia Badene, Rudy Baraglia, Yazid Benazzouz, Sami Benhamiche, Abdel Wahab Heba, Yoann Houpert, Tom Jorquera, Damien Lainé, Romain Lopez, Michel-Marie Maudet, Ilyes Rebai, Zied Sellami, Samir Tanfous, Kate Thompson, Alexandre Zapolsky, and Sarah Zribi.

Finally, I would like to heartily thank my family and friends, Xiaohong Tao, Tingbang Shang, Wenjie Shang, Qiumei Tao, Pan Jia, Gengfei Fan, Yunpeng Fei, Zhenyu Yao, Wei Zhong, Ruiqing Yin, Jie Lu, and Jiyuan Liu, for their warm support throughout all these years of a long, tough but amazing student trip.

Thanks to all people, that my memory allows me to include here or not, who have contributed in their own way towards the completion of this thesis.

Guokan Shang
Guyancourt, February 2020

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Research in *meeting analysis* (Romano and Nunamaker, 2001)—the study of meeting expenses, productivity, processes, and outcomes—has revealed that while meetings are an essential and inevitable part of one's working life on the one hand, they are sometimes considered to be costly, unproductive, and dissatisfying on the other. Living in the booming era of Artificial Intelligence (AI), our lives are constantly being affected by the increasing deployment of AI-based systems in a wide variety of contexts (Lu, 2019), from conversational agents to autonomous driving, to the healthcare industry. The time is ripe for an AI-powered assistant that intelligently understands various aspects of meetings, helps us be more effective during our discussions, and delivers reliable records of meetings afterwards.

1.1 SPOKEN LANGUAGE UNDERSTANDING

In order to teach machines to understand speech, the medium through which people most naturally interact, researchers developed the field of *Spoken Language Understanding* (SLU) (Tur and De Mori, 2011), which lies between the fields of speech processing and language processing and aims at investigating conversational speech by leveraging advances in Automatic Speech Recognition (ASR), Natural Language Processing (NLP), and Machine Learning (ML).

In this context, the meeting assistant can be seen as one type of SLU system for processing human-human conversation. Once the audio recording of a meeting is processed by an automatic speech recognizer that translates speech to text, the meeting assistant uses the resulting transcription as input to various downstream SLU components, such as:

DIALOGUE ACT SEGMENTATION AND TAGGING, in which an unstructured stream of words is divided into sentential units called *utterances* or *dialogue acts*, and then each utterance is assigned a dialogue act label representing the underlying communicative intention it conveys (e.g., stating, questioning, answering, etc.).

TOPIC SEGMENTATION AND IDENTIFICATION, in which a conversation is divided into topically coherent sets of utterances, and each set is assigned a topic.

REFERENCE AND ADDRESSEE RESOLUTION, which aims to determine to what or to whom the speaker is speaking, listening, or referring.

ACTION ITEM AND DECISION DETECTION, whose goal is to detect relevant utterances/areas of discussion with regard to task-assignment and decision-making.

SUMMARIZATION, which aims to generate a shortened version of the meeting discussion while keeping important points.

Much effort has been devoted to the development of meeting assistants through various research projects. One of the earliest contributions was that of the CALO meeting assistant (Tur et al., 2008, 2010), which is an automatic agent that provides for distributed meeting capture, annotation, automatic transcription and semantic analysis of multiparty meetings. The more recent REUs project (Alizadeh et al., 2018; Jacquenet, Bernard, and Largeron, 2019; Doan et al., 2020) aims to provide management tools for meetings and to automatically generate meeting minutes based on the audio signal provided by lightweight materials, such as omnipresent smartphones and personal computers, rather than meeting rooms equipped with high quality sound systems. Meeting assistants also attract interest from the industrial sector. The principal goal of the LinTO project (Lorré et al., 2019; Rebai et al., 2020), for instance, is to develop such an open source assistant for professional use in a corporate environment. The LinTO assistant is designed to be deployed either as a platform or as a physical interactive device equipped with microphones, a screen, and a 360° camera. LinTO has a highly configurable client-server architecture, and through a user-friendly console the provided components can be used to design, build and mange customized assistants. The functionalities, described as skills (e.g., ASR and NLP tasks), can be easily added to the manipulable SLU workflow. In addition, LinTO can be deployed in any OS system and handle a high number of queries, thanks to the Docker technology (Merkel, 2014) it relies on.

The work described in this dissertation, developed as a CIFRE thesis[1] in the context of the LinTO project and in collaboration with the industrial partner LINAGORA, contributes to ongoing efforts to teach machines to understand multi-party meeting speech, and more specifically, to automatically generate meeting summaries. Analogous to human meeting minutes, such documents would be of great value to meeting participants and non-participants alike (e.g., managers and auditors), and for individuals as well as organizations.

1.2 MEETING SUMMARIZATION

Automatic Text Summarization (Gambhir and Gupta, 2017) is the task of distilling the key informational elements from a text and using them to produce a significantly condensed version of the original text.

1. *Conventions Industrielles de Formation par la REcherche* (CIFRE) is an industrial Ph.D. program in France supported by the *Association Nationale Recherche Technologie* (ANRT).

Summarization approaches fall into two broad categories: *extractive* and *abstractive*. The former creates summaries by directly lifting important sentences from source documents, without any further modifications. The latter involves the generation of more human-like and novel abstractive sentences based on a deeper understanding of the meaning of the original text, via a Natural Language Generation (NLG) component. Table 1.1 shows an example of both an extractive and abstractive summary of the same dialogue.

extractive summary:
UI: But what if we ha what if we had like a Spongy sort of like stress balley kinda [disfmarker]
PM: If you have like that stress ball material kind of as what you're actually holding in your hand,
ME: Because I was thinking if you have a cover for the squashy bit,
UI: oh so so you're saying the squishy part's like detachable,
UI: so so maybe one you know [disfmarker] you can have like the broccoli squishy thing, and then you could have like the banana squishy thing
…
abstractive summary:
Some part of the casing will be made of a spongy material.
…

Table 1.1 – A comparison of an extractive and abstractive summary of the ES2011c AMI meeting (McCowan et al., 2005).

In research on automatic summarization, the extractive approach has received far more attention than the abstractive approach, mainly because abstraction is a more complex and challenging process than extraction, as it requires the generation of novel sentences based on a higher level of linguistic understanding. Human judges generally prefer abstractive summaries, however, especially for text made up of dialogues (Murray, Carenini, and Ng, 2010). Indeed, extractive summaries are ill-suited for speech, in which information is often spread across multiple turns in a dialogue. Cutting and pasting mere fragments of such exchanges, taken out of their original contexts, renders extractive summaries of spoken conversation largely unintelligible. In this thesis, we focus on the abstractive summarization approach.

It is widely recognized that reliable automatic summarization methods would be immensely useful for coping with the well-known information overload problem with which we are confronted on a daily basis (Edmunds and Morris, 2000). Yet, while summarization for traditional textual documents (e.g., news) is an extensively-studied topic, summarization of multi-party conversations (Murray, 2008; Carenini, Murray, and Ng, 2011) remains a comparably emerging and underdeveloped research area, even if it has recently been gaining attention.

This asymmetry is due in large part to the nature of multi-party conversation, which poses challenges not encountered with traditional text, but also a lack of data and appropriate evaluation metrics. Such problems force us to adopt novel methods that move well beyond the state of the art for text summarization.

DATASETS. Thanks to the impressive progress of automatic speech recognition technology and its ubiquitous adoption in web conferencing tools, vast amounts of high quality automatic speech transcriptions are becoming increasingly available. However, very few published datasets are annotated with summaries for research purposes. There are only two available meeting datasets in English (the AMI and ICSI corpora; Janin et al., 2003; McCowan et al., 2005), and there are no such resources for other languages. As supervised learning methods often need large amounts of training data, and trained models are usually language-dependent and domain-dependent, one way to get around the data scarcity problem might be to move to unsupervised summarization techniques.

MEETING AND SUMMARY TYPES. Because there are many different types of meetings carried out for different purposes (Nedoluzhko and Bojar, 2019), there is arguably no one-size-fits all approach to summarization. Focused summaries (Fernández et al., 2008; Bui et al., 2009; Wang and Cardie, 2011, 2012, 2013) that provide an outline of proposed ideas, supporting arguments, and decisions might be more appropriate for decision-making meetings, for example, while general topic summaries might be better suited for information-sharing meetings, and template-based summaries (Oya et al., 2014), for well-framed meetings with clear agendas. Given that summaries may vary with respect to their format, style, and content, approaches to automating production will arguably need to be conducted with a variety of methods to allow a system to produce a summary type tailored to a particular meeting format.

NATURE OF SPEECH AND NOISE. Unlike well-formed text documents, spontaneous conversation consists of complex multi-party interactions with extreme variability. The language is informal with low information density, and utterances tend to be partial, fragmentary, ungrammatical, overlapping, and include many disfluencies, ellipses, and pronouns. ASR transcription and segmentation errors inject additional noise into the input, especially under non-ideal recording conditions. One needs to take into account these inherent differences of speech from text (Zechner, 2002) when developing speech summarization techniques. One way to equalize the hurdles introduced by spoken conversation might be to incorporate information from the acoustic signal, such as the nature of pauses or prosodic features, to improve performance

over a system that draws on lexical features alone (Xie and Liu, 2010).

EVALUATION. Not only for meetings but also for broader text summarization, evaluation is itself a challenging task. For one thing, there is normally no single best summary for a given source document. Different annotators can create very different but equally valid summaries. In fact, Rath, Resnick, and Savage (1961) showed that even for a single person asked to summarize the same document twice several weeks apart, the outcomes can be very different. Another problem is that the ROUGE metric, the standard nowadays for summary evaluation (Lin, 2004), is biased towards lexical similarities: evaluation is conducted based on surface lexicographic matches, making it unsuitable for assessing the quality of generated abstractive summaries that are semantically but not lexically similar to human summaries. Nor does ROUGE take into account the readability or fluency of the generated summaries (Ng and Abrecht, 2015). Given the special characteristics of dialogues, more studies will be needed to develop more appropriate evaluation metrics for speech summarization (Zechner and Waibel, 2000).

The recent explosion of research and development in *Deep Learning* (DL) (Goodfellow et al., 2016), especially the invention of sequence-to-sequence (encoder-decoder) (Sutskever, Vinyals, and Le, 2014) architectures enhanced with attention mechanisms (Bahdanau, Cho, and Bengio, 2014; Vaswani et al., 2017), has helped to propel research on abstractive summarization forward. We have seen performance boosts from applying DL to traditional text documents (Rush, Chopra, and Weston, 2015; Nallapati et al., 2016; See, Liu, and Manning, 2017a; Lewis et al., 2019) and on documents from the meeting domain (Li et al., 2019a; Zhu et al., 2020). Nevertheless, we are far from being perfect and producing commercial-level automatic tools that generate abstractive summaries of human-level quality. The task remains one of the most challenging NLP tasks, with many open problems to be resolved (Kryscinski et al., 2019).

1.3 THESIS STATEMENT

This dissertation contributes pipelines, models, components, and new insights to problems that arise in the area of spoken language understating/meeting summarization. In particular, we have developed:

— A fully unsupervised framework based on multi-sentence compression graphs and budgeted submodular maximization for abstractive meeting summarization: this approach takes a speech transcription as input and generates a summary.

— A modified neural conditional random field layer that takes speaker-change into account for dialogue act classification: this approach assigns each utterance a dialogue act label to represent its communicative intention.

— An energy-based learning approach, a general triplet sampling scheme, and a contextual utterance encoder featuring self-attention mechanisms for abstractive community detection: this method groups utterances in a conversation according to whether they can be jointly summarized by a common abstractive sentence.

We elaborate on these three contributions below.

1.4 OVERVIEW OF CONTRIBUTIONS

Our research has been conducted to address three specific tasks, which remain open problems in the domain of SLU: abstractive meeting summarization, dialogue act classification, and abstractive community detection. We consider the latter two tasks as stepping stones towards generating better abstractive summaries. In more detail, our work has been motivated by, and has contributed to answering, the following research questions:

ABSTRACTIVE MEETING SUMMARIZATION.
How can we generate abstractive meeting summaries in an unsupervised way?

A significant issue for research in meeting summarization is the lack of annotated data. As explained above, meeting transcriptions are becoming increasingly available, but few are accompanied with abstractive summaries for research purposes due to the enormous human effort and cost required for annotating. Supervised summarization models often need to be fed with large amounts of training data in order to learn from human-labelled examples. Moreover, the resulting models are usually language-dependent and domain-dependent. These concerns ultimately drove us to investigate unsupervised summarization techniques.

In Chapter 3 of the thesis, we introduce a novel graph-based framework for abstractive meeting speech summarization that is fully unsupervised, does not rely on any annotations, and can be applied to any languages other than English in an almost out-of-the-box fashion. Within our framework, transcriptions are successively processed through 4 modules:

1. The first module preprocesses text.
2. The second module groups together the utterances that should be summarized by a common abstractive sentence, for instance, with respect to a topic or subtopic.

3. The third module generates a single abstractive sentence for each utterance group, where we present an unsupervised NLG component: a Multi-Sentence Compression Graph (MSCG) extended with novel edge and path weighting schemes. Our extended MSCG can yield a better abstractive sentence that is more fluent, informative, and diverse than the original MSCG of Filippova (2010). To accomplish this, we combine the strengths of multiple recent approaches (while addressing their weaknesses) from different NLP tasks, and leverage advances in word embeddings, graph-of-words, and graph degeneracy.
4. The goal of the last module is to generate a summary by selecting the best elements from the NLG module's output under a budget constraint (summary size). We cast this problem as the maximization of a custom submodular quality function.

We show via experiments that our system improves on the state-of-the-art and generates reasonably grammatical abstractive summaries despite taking noisy utterances as input and not relying on any annotations.

DIALOGUE ACT CLASSIFICATION.
Is speaker information useful for dialogue act classification?

Although our unsupervised approach can generate readable abstracts of meetings, the quality is still far from those produced by humans, and they are somewhat extractive and spoken-style. The reason is that the MSCG is restricted to formulate a new sentence by simply compressing multiple original utterances, thus cannot freely bring novel words as expressive as humans do. We believe that to make summaries more abstractive and written-style, we need first to understand meeting speech deeply. In other words, the systems need to be incorporated with more knowledge about the discourse, such as the meanings of utterances and their relationships, topics, etc. Building upon a rich understanding of conversations will enable unsupervised/supervised meeting summarization techniques to go one step further in performance.

To this end, we investigate the problem of dialogue act (DA) classification, which aims to assign each utterance a DA label to represent its communicative intention, such as suggestion, question, agreement, and so on. DAs provide a preliminary understanding of speakers' intentions and serve as helpful annotations for a large variety of downstream conversation processing tasks. DAs have been proven useful in the study of focused meeting summarization (Wang and Cardie, 2012).

Recent work in DA classification approaches the task as a sequence labeling problem, using neural network models coupled with a Conditional Random Field (CRF) as the last layer. CRF models the con-

ditional probability of the target DA label sequence given the input utterance sequence. However, the task involves another important input sequence, that of speakers, which is ignored by previous work. In Chapter 4 of this thesis, we propose a simple modification of the CRF layer that takes speaker information into account to address this limitation. More specifically, in our modified CRF layer, the label transition matrix is conditioned on speaker-change. Our contribution is general, since the layer can be plugged on top of any deep learning component to form a DA classification model. We evaluate it within the BiLSTM-CRF architecture, experiments show that our modified CRF layer outperforms the original one, with very wide margins for some DA labels. Further, visualizations demonstrate that our CRF layer can learn meaningful, sophisticated transition patterns between DA label pairs conditioned on speaker-change in an end-to-end way.

We can conclude that taking speaker information into consideration is beneficial to the task of DA classification.

ABSTRACTIVE COMMUNITY DETECTION.
What is the connection between extractive summarization and abstractive summarization?

In Chapter 5 of the thesis, we rethink the general process of abstractive summary generation to make it more consistent with human behavior. To this end, we look into how relevant annotations were initially created in the AMI and ICSI meeting corpora. As described in the annotation guideline: for each meeting with its manual transcription, annotators were asked to 1) write an abstractive summary, 2) extract important utterances as an extractive summary, 3) link extracted utterances with the sentences written in the abstractive summary. All utterances linked with a same abstractive sentence form a group called an *abstractive community*, which is the source of information based on which the abstractive summary sentence was written. Table 1.1 shows an example, where all utterances in the extractive summary are linked to the sentence in the abstractive summary.

Inspired by the above annotating process, we argue that the abstractive meeting summarization task needs to be done in three steps: 1) find a set of important utterances, 2) group selected utterances into abstractive communities, 3) generate an abstractive sentence for each of the communities. This pipeline matches the general and basic summarization process described in the work of Jones et al. (1999), as consisting of Interpretation, Transformation, and Generation:

1. Interpretation. Mapping the input text to a source representation.
2. Transformation. Transforming the source representation to a summary representation.

3. Generation. Generating a summary text from the summary representation.

We note that the first and the third steps have been extensively studied as extractive summarization and NLG, while the second step is a task little explored since its introduction by Murray, Carenini, and Ng (2012) under the name of Abstractive Community Detection. We think this is an important problem, as it plays the crucial role of bridge between extractive and abstractive summarization.

We introduce a novel approach to this task, which is essentially an utterance clustering problem. We first present a neural contextual utterance encoder featuring three types of self-attention mechanisms. We then train it using the siamese and triplet energy-based architectures (LeCun and Huang, 2005; Lecun et al., 2006) from the area of deep metric learning (Hoffer and Ailon, 2015; Mueller and Thyagarajan, 2016), whose objective is to project training utterances into an embedding space in which the utterances from a given abstractive community are close to each other. For the triplet architecture, we also propose a general sampling scheme that enables the architecture to capture subtle clustering patterns, such as overlapping and nested abstractive communities. Finally, we apply the Fuzzy c-Means clustering algorithm on the trained utterance embeddings in order to obtain abstractive communities. Experiments and visualization show that our system outperforms multiple energy-based and non-energy based baselines from the state-of-the-art and that our triplet sampling scheme is effective.

1.5 SOFTWARE AND LIBRARIES

The primary software and libraries that were used in the context of this thesis are the following:

— `Scikit-learn` (Pedregosa et al., 2011). A machine learning library for the Python programming language.

— `Keras` (Chollet et al., 2015). A deep learning API written in Python, running on top of the machine learning platform TensorFlow (Abadi et al., 2015).

— `Gensim` (Řehůřek and Sojka, 2010a). A library for topic modeling, document indexing, and similarity retrieval with large corpora.

— `Numpy` (Oliphant, 2006; Van Der Walt, Colbert, and Varoquaux, 2011). A fundamental package for scientific computing in Python.

— `Networkx` (Hagberg, Schult, and Swart, 2008). A Python package for creating, manipulating, and studying the structure, dynamics, and functions of complex networks.

— `Igraph` (Csardi, Nepusz, et al., 2006). A library for creating, manipulating, and analyzing graphs.

— `Matplotlib` (Hunter, 2007). A library for creating static, animated, and interactive visualizations in Python.

— `ROUGE 2.0` (Ganesan, 2015). An evaluation toolkit for Automatic Summarization tasks.

Note that we made all source code and preprocessed data publicly available for reproducibility and for fostering research on the topics covered by this thesis.

1.6 OUTLINE OF THE THESIS

The rest of the dissertation is organized as follows. Chapter 2 provides some cornerstones upon which our work was built in terms of text representation techniques, metrics to evaluate our developed approaches, and descriptions of datasets on which we conducted the experiments. The next three chapters are devoted to presenting three primary tasks under study in this thesis, the related work in the literature, our motivation, and our contributions over the state-of-the-art. In particular, Chapter 3 presents our unsupervised approach for abstractive meeting summarization, Chapter 4 , our contributions to dialogue act classification, and Chapter 5, our work on abstractive community detection. Finally, Chapter 6 concludes the dissertation and sheds some new light on future research directions.

2 BASIC CONCEPTS AND PRELIMINARIES

In this chapter, we provide some basic concepts and the minimum background required to follow the rest of the thesis. We discuss in detail: 1) the text representation techniques used in our work so that machines can process natural language, 2) the evaluation metrics adopted in our experiments to measure the performances of our developed approaches, 3) the data with relevant annotations on which we report the results. For a deeper understanding of these areas, one may need to refer to books on Machine Learning (Bishop, 2016), Deep Learning (Goodfellow et al., 2016), Natural Language Processing (Manning and Schutze, 1999; Jurafsky et al., 2014), Spoken Language Understanding (Huang et al., 2001; Tur and De Mori, 2011), and Conversation Summarization (Carenini, Murray, and Ng, 2011).

2.1 TEXT REPRESENTATION

Machines understand the world only through mathematical representation, i.e., numbers rather than information in its original form. The central objective of NLP is to determine how we can effectively and efficiently convert natural language in textual form to machine-readable representation. In this section, we present three different text representation techniques used in our research to enable machines to process text.

The underlying structure of text is hierarchical. Characters combine into words, words combine into sentences, and sentences combine to form documents. Documents can then be assembled into sets called *collections*. Different NLP tasks study problems at different levels of granularity; in our research, we are mainly interested in word-level and sentence-level representation. From now on, we use t, d, D to denote a word (term), a sentence, and a document, respectively. Note that their denotations may vary depending on the specific task at hand.

2.1.1 *Bag-of-words*

Given a vocabulary V (a finite list of indexed words), a word t can then be represented as a one-hot encoded vector $\mathbf{x}_t$ of size $|V|$, in which the element corresponding to the vocabulary index of t is one and all the other elements are zero. A sentence d can then be represented as a

weighted sum of the vectors for all the words contained in the sentence, as follows:

$$\mathbf{x}_d = \sum_{t \in d, d \in D} f(t, d, D)\mathbf{x}_t \tag{2.1}$$

where $f(t, d, D)$ is a weighting function for the word t in its context of d and D. The simplest approach is to set $f(t, d, D)$ to be equal to the number of occurrences of t in d, which is referred to as the *Term Frequency* weighting function $TF(t, d)$ (Luhn, 1957). The *importance* (weight) of t in d is thus simply proportional to its frequency.

This text representation technique is known in the literature as the Bag-of-Words (BoW) model (Manning, Schütze, and Raghavan, 2008) because it represents text as if it were a bag of words, i.e., an unordered set of words that does not keep track of where a word appeared in a sentence d or document D. Despite the loss of information engendered by the assumption of *term independence*, i.e., that the presence of one word in the bag is independent of another, BoW can be surprisingly effective on some tasks and is extensively used as a simple baseline.

Apart from term frequency, a more widely-used and thoughtful term weighing function is TF-IDF (Salton and Buckley, 1988), where IDF stands for *Inverse Document Frequency* (Jones, 1972). It is defined as follows:

$$TF\text{-}IDF(t, d, D) = TF(t, d) \times IDF(t, D) \tag{2.2}$$

$$IDF(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{2.3}$$

where $|D|$ is the number of sentences and $|\{d \in D : t \in d\}|$ is the number of sentences in D containing t. The intuition behind TF-IDF is that a word should be considered important within a given sentence if it has a high term frequency *and* if the number of sentences in which the word appears is relatively small. With applying IDF, the weights of the words occurred too often in many different sentences are thus decreased, giving more importance to the more specific words that better discriminate between sentences.

The primary limitation of BoW is its inability to capture syntactic and semantic relations between words. For example, the words `"bike"` and `"bicycle"` are distinct under the view of BoW (e.g., the cosine similarity between their vectors is zero), even though they are identical in terms of semantic meaning and syntactic function. Moreover, since BoW discards the order of words, which is very important in determining sentential meaning, the sentence `"Beijing is the capital of China."` is treated as identical to `"China is the capital of Beijing."`. Furthermore, output vectors are inherently high-dimensional and sparse due to large vocabulary size, which can lead to the *curse of dimensionality* (Bellman, Corporation, and Collection, 1957; Bellman and Collection, 1961) for machine learning models.

Note that throughout this chapter, we focus on discussing the most popular TF-IDF weighting function for the BoW model, which creates orthogonal vectors for all words (`"bike"` vs. `"bicycle"`). The other weighting functions developed in the literature of distributional semantic models, such as the Positive Pointwise Mutual Information (PPMI) weighting function (Evert, 2005; Baroni, Dinu, and Kruszewski, 2014) for the BoW model can actually creates similar vectors for similar words (i.e., capturing lexical semantic similarity). Interested readers are referred to the work of Turney and Pantel (2010) for details.

In the next two subsections, we present two alternative text representation techniques, one of which overcomes some drawbacks of BoW and the other of which overcomes all.

2.1.2 *Graph-of-words*

The Graph-of-Words (GoW) model or word co-occurrence network (Mihalcea and Tarau, 2004; Rousseau and Vazirgiannis, 2013) is a graph-based text representation technique. In GoW, a sentence/a piece of text is modelled as a graph whose vertices represent unique terms and whose edges correspond to co-occurrence relationships between the terms within a fixed-size sliding window over the text.

For a given sentence, e.g. `"information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources"`, we can create a GoW as illustrated in Figure 2.1. An interactive web application GoWvis (Tixier, Skianis, and Vazirgiannis, 2016) that illustrates the GoW model is available online: `https://safetyapp.shinyapps.io/GoWvis`.

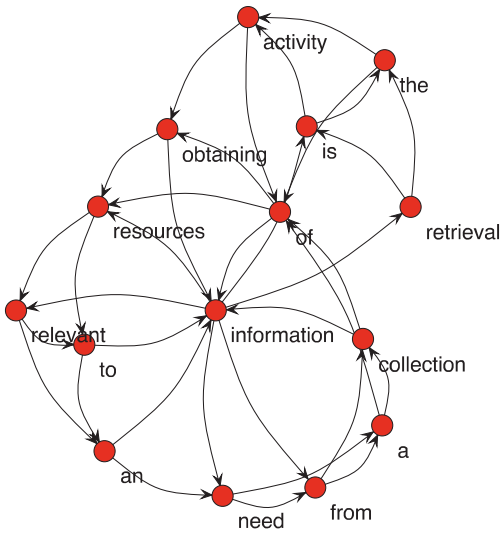

Figure 2.1 – Example (Rousseau and Vazirgiannis, 2013) of an unweighted directed GoW in which an edge indicates at least one directed co-occurence of the two terms in a window of size 3 in the text.

There are many variants of GoW in terms of edge directionality and edge weighting. Moreover, changing preprocessing steps or window size may significantly influence the graphs produced and, consequently, the results for certain tasks. Further discussion of these points falls out of the scope of this dissertation, but we refer the reader to the work of Rousseau (2015) for a more extensive review of the subject.

Similar to TF-IDF, we can derive TW-IDF for GoW as follows:

$$TW\text{-}IDF(t, d, D) = TW(t, d) \times IDF(t, D) \tag{2.4}$$

where the *Term Weight* weighting function $TW(t, d)$ can be any centrality measures (e.g., degree, closeness, and betweenness (Easley, Kleinberg, et al., 2010; Newman, 2010)) that determine the relative importance of a node(term) t in the graph representation G_d of the sentence d. The intuition here is if a node is important in the graph (this is, located in the network center), then its corresponding term is equivalently important in the original text.

The primary advantage of GoW over BoW is that it enables graph theory to be applied to text and does not make the term independence assumption when constructing graphs, so that we can exploit more information about word dependence, order, and distance. This text representation technique has already been successfully applied to various tasks, including information retrieval (Rousseau and Vazirgiannis, 2013), keyword extraction (Rousseau and Vazirgiannis, 2015; Tixier, Malliaros, and Vazirgiannis, 2016; Meladianos et al., 2017), extractive summarization (Tixier, Meladianos, and Vazirgiannis, 2017), subevent detection (Meladianos et al., 2015), and representation learning (Nikolentzos, Tixier, and Vazirgiannis, 2019).

2.1.3 *Dense vector representation in deep learning*

Both BoW (with TF-IDF) and GoW (with TW-IDF) represent a word or a sentence as a *sparse, long* vector in a space whose dimensions are the unique words in the vocabulary. While GoW avoids some drawbacks of BoW mentioned earlier, e.g., by taking word order into account when constructing graphs, it still fails to encode semantic similarity between words. To remedy this problem, it is better to use a *short, dense* vector so that its dimensions represent higher-level underlying aspects of word meaning (e.g., singular/plural, positive/negative, male/female, biological/artificial), yielding similar vectors for similar words (e.g., `"bike"` and `"bicycle"`).

Mikolov et al. (2013a,b) introduce continuous bag-of-words (CBOW) and continuous skip-gram models to learn such representations of word meaning—known as *word embeddings* (as words are embedded in a vector space)—directly from their distributions in unannotated text. The CBOW model predicts a target word based on the embeddings of its context words, while the skip-gram model predicts surrounding

words given the embedding of a target word. These models can be seen as instantiations of the *distributional hypothesis* of language (Joos, 1950; Harris, 1954; Firth, 1957), which states that words that occur in similar contexts tend to have similar meanings (or functions). This hypothesis suggests that given a new word, one should be able to figure out its meaning based on the contexts in which it is used. Conversely, when confronted with a word whose meaning is known, one should be able to predict the contexts in which that word is likely to occur; that is, the words with which it is likely to be combined to make a sentence.

CBOW and skip-gram models are very similar in design, so we only present the latter in detail. The skip-gram model turns the problem of learning representations into a binary classification task, that of answering a set of questions of the form, *"Is a word t likely to show up near a word c?"*. We create a training set that consists of positive pairs $(t, c) \in +$ and negative pairs $(t, c) \in -$ from collected text examples. When (t, c) is positive it means c occurs near t within a given window (preceding and following) in the text. In the negative case, c can be any other word randomly sampled from the vocabulary following a *negative sampling* strategy. The training objective consists in 1) maximizing the probability P(+|t,c) that c is a real context word of t, for all positive pairs $(t, c) \in +$, and 2) maximizing the probability P(-|t,c) that c is *not* a real context word of t, for all negative pairs $(t, c) \in -$. As described below:

$$L(\theta) = \sum_{(t,c)\in+} logP(+|t, c) + \sum_{(t,c)\in-} logP(-|t, c) \tag{2.5}$$

$$P(-|t, c) = 1 - P(+|t, c) \tag{2.6}$$

$$P(+|t, c) = \sigma(\mathbf{x}_t \cdot \mathbf{x}'_c) \tag{2.7}$$

where σ denotes the logistic (sigmoid) function, which turns the dot product of word embeddings $\mathbf{x}_t \cdot \mathbf{x}'_c$ into a probability value. We can see that the model bases this probability on dot product similarity, i.e. we want to maximize the similarity of the target word with those words that occur nearby in texts, and minimize the similarity of the target word with those words that don't occur nearby, which reflects the distributional hypothesis: *"a word is characterized by the company it keeps."* (Firth, 1957).

Note that since word embeddings are automatically learned rather than hand-crafted, the dimensions are usually not interpretable by humans, and specific dimensions do not necessarily correspond to specific concepts. Nevertheless, researchers have found some interesting semantic properties. For example, Mikolov, Yih, and Zweig (2013) show that the offsets between embeddings can capture some analogical relations between words. A well-known example is that the result of the expression $\mathbf{x}_{\texttt{king}} - \mathbf{x}_{\texttt{man}} + \mathbf{x}_{\texttt{woman}}$ is a vector close to $\mathbf{x}_{\texttt{queen}}$.

As a result of these advantages, word embeddings are now the most popular way to represent the meaning of words in NLP. Moreover,

they are well suited for neural networks, since they can be directly plugged in as inputs to subsequent neural components. In the work of this dissertation, for instance, we mainly use LSTMs (Hochreiter and Schmidhuber, 1997) and GRUs (Cho et al., 2014), both of which are variants of Recurrent Neural Networks (RNNs) that basically map a sequence of inputs (word embeddings of the words in a sentence) $\{\mathbf{x}_1 \dots \mathbf{x}_T\}$ to a sequence of outputs $\{\mathbf{y}_1 \dots \mathbf{y}_T\}$, as well as a sequence of hidden states $\{\mathbf{h}_1 \dots \mathbf{h}_T\}$ in a recursive manner, as described below:

$$\mathbf{h}_t = g(\mathbf{U}\mathbf{h}_{t-1} + \mathbf{W}\mathbf{x}_t) \tag{2.8}$$

$$\mathbf{y}_t = f(\mathbf{V}\mathbf{h}_t) \tag{2.9}$$

where f and g are activation functions (e.g., tanh, sigmoid, softmax) and $\mathbf{U}, \mathbf{W}, \mathbf{V}$ are weight matrices, shared across time, to be trained with respect to a loss function, an optimizer, a dataset. The $\mathbf{y}_T$, the output vector of the last timestep T (corresponding to the last word of the sentence), thus can be seen as a sentence representation.

2.2 EVALUATION

In this section, we present the evaluation metrics used to measure the performance of the different approaches developed in this dissertation.

2.2.1 *Accuracy, Precision, Recall, and F1-score*

Given a binary classification task with examples and their ground truth positive/negative labels, there are four types of possible predictions (Swets, 1963) made by a system, as summarized in the *confusion matrix* in Table 2.1: True Positives (TP, correctly classified as positive), True Negatives (TN, correctly classified as negative), False Positives (FP, incorrectly classified as positive), and False Negatives (FN, incorrectly classified as negative).

		Predicted class	
		positive	negative
Actual class	positive	TP	FN
	negative	FP	TN

Table 2.1 – Confusion matrix for binary classification.

Note that there are TP+FN positive and FP+TN negative ground truth examples in total, and there are TP+FP and FN+TN examples predicted as positive and negative by the system respectively.

Given the confusion matrix, the *accuracy* is defined as:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where the numerator is the total number of correct predictions, and the denominator is the number of all available examples. Accuracy evaluates the overall fraction of predictions that are correct.

The Precision (P), Recall (R), and F1-score ($F1$) (Allen et al., 1955; Van Rijsbergen, 1979) are calculated as follows:

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN} \quad F1 = \frac{2 \times P \times R}{P + R} \tag{2.10}$$

P evaluates the fraction of examples predicted positive that are truly correct. R evaluates the fraction of truly correct examples that are predicted positive. $F1$ is a measure that trades off Precision versus Recall, which is the Harmonic mean of them.

These metrics can be easily adapted to other tasks in some way. For example, information retrieval systems aim to find relevant documents from a collection that best match a given query sentence. In this context, TP can be defined as *relevant documents retrieved*, and Precision thus evaluates the fraction of retrieved documents that are relevant. Moreover, in the case when we are only interested in evaluating the quality of the highest ranked documents retrieved by a system, we can keep top k results and drop the others for calculation, leading to the P, R, and $F1$ at k metrics.

2.2.2 *ROUGE*

ROUGE (Lin, 2004) is the most popular measure for summarization evaluation, which can estimate the quality of a candidate (system-generated) summary against (one or multiple) reference (human) summaries. ROUGE is made up of several metrics based on *comparing n-gram overlap*, i.e., ROUGE-1 considers unigrams, ROUGE-2 considers bigrams, ROUGE-SU4 considers unigrams plus skip-bigrams with a maximum skip distance of 4. ROUGE can also be seen as an adaptation of Precision, Recall, and F1-score. For example, ROUGE-1 Recall is calculated as follows:

$$\text{ROUGE-1 R} = \frac{number\_of\_overlapping\_words}{total\_words\_in\_reference\_summary} \tag{2.11}$$

It can be interpreted as: the more that the unigrams from the reference summary also appear in the candidate summary, the higher the Recall score is.

In the case when there are multiple available reference summaries, ROUGE simply pairwisely computes scores between the candidate and one of reference summaries and outputs the maximum score.

2.2.3 *Omega index*

The Omega index (Collins and Dent, 1988) 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 Equation 2.12. 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.

$$\omega(s_1, s_2) = \frac{\omega_{obs}(s_1, s_2) - \omega_{exp}(s_1, s_2)}{1 - \omega_{exp}(s_1, s_2)} \tag{2.12}$$

Observed and expected agreements are calculated as below:

$$\omega_{obs}(s_1, s_2) = \frac{1}{N_{total}} \sum_{j=0}^{min(J,K)} A_j \tag{2.13}$$

$$\omega_{exp}(s_1, s_2) = \frac{1}{N_{total}^2} \sum_{j=0}^{min(J,K)} N_{j1} N_{j2} \tag{2.14}$$

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:

$$s_1 = \{\{a, b, c\}, \{b, c, d\}, \{c, d, e\}, \{c, d\}\}$$
$$s_2 = \{\{a, b, c, d\}, \{b, c, d, e\}\}$$

	solution s_1	solution s_2	solutions
	#communities the pair is assigned to	#communities the pair is assigned to	s_1 and s_2 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 2.2 – Omega index example.

Solutions are transformed into Table 2.2, from which 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, the Omega index for this simple example is computed as: $\omega(s_1, s_2) = (0.6 - 0.36)/(1 - 0.36) = 0.375$.

2.3 DESCRIPTION OF DATASETS

In the area of meeting summarization, the AMI and ICSI corpora are both widely-used and in fact, the only-available English meeting corpora. Not only do they come with summaries for each meeting, but they are designed to support the work of several research communities with many other resources (e.g., audio and video recordings) and annotations.

THE AMI CORPUS: (McCowan et al., 2005) contains 137 scenario-driven meetings (65 hours) recorded under the Augmented Multi-party Interaction project. In each meeting ranging from 15 to 45 minutes, four participants play the roles of a project manager (PM), a marketing expert (ME), a user interface designer (UI), and an industrial designer(ID) within a fictive electronics company. The scenario given to them is that they form a design team, whose task is to develop a new television remote control from inception to market, through individual work and a series of group meetings. Note that even though the scenario is artificial, the participants' speech is spontaneous and the interaction is natural.

THE ICSI CORPUS: (Janin et al., 2003) consists of 75 naturally-occurring meetings (72 hours) recorded at the International Computer Science Institute. In each meeting of around 1 hour (regularly scheduled weekly), members from research groups (e.g., undergraduate student, Ph.D. student, and professor) discuss specialized and technical topics such as natural language processing, neural theories of language, ICSI corpus related issues. There are 6 participants on average per meeting.

To produce the meeting summaries, human annotators were given instructions on how to create both abstractive and extractive summaries. Below, we present the four types of annotations that were used to conduct our experiments: speech transcription, extractive summarization, abstractive summarization, and abstractive-extractive linking. We visually illustrate the relationships between these annotations in Figure 2.2 by taking as an example an entire meeting from the AMI corpus.

A more detailed version of this Figure can be found in Figure 5.2 of Chapter 5.

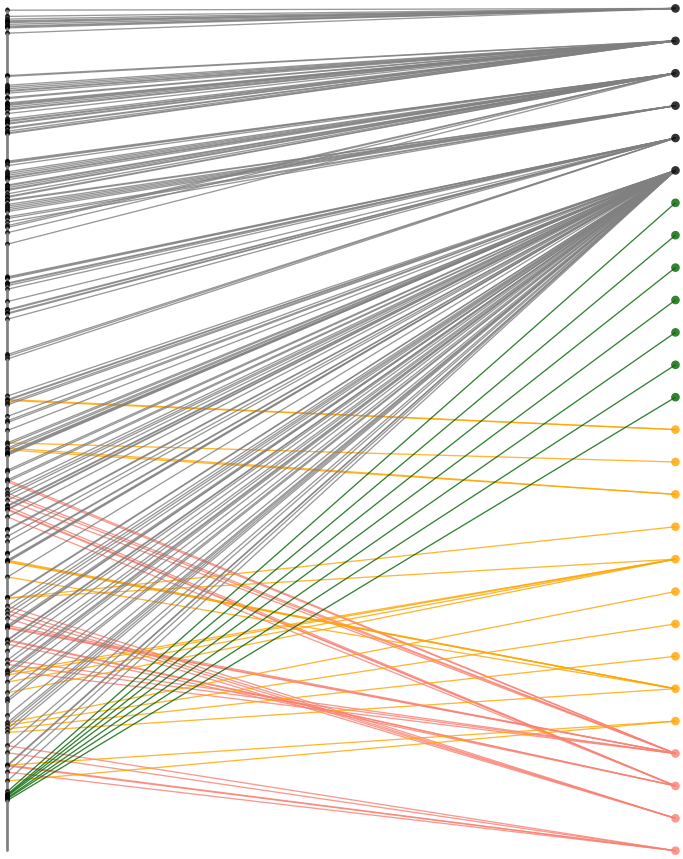

Figure 2.2 – Example of ground truth human annotations from the ES2008b AMI meeting. Successive grey nodes (displayed as a straight line) on the left denote utterances in the transcription, where black nodes correspond to extractive summary, i.e., the utterances judged important (summary-worthy). Each of the nodes on the right represents a sentence from the abstractive summary. All utterances linked to the same abstractive sentence form one abstractive community.

SPEECH TRANSCRIPTION: A sequence of utterances over time, each utterance represents a segment of speech by a single speaker. Utterances are associated with extra attributes, including start/end timestamp, speaker, dialogue act, addressee, etc. Transcriptions are either manual, created by an annotator, or automatic, generated by an ASR system. In Figure 2.2, successive utterances of the entire transcription are represented as the grey vertical line of nodes on the left.

EXTRACTIVE SUMMARY: A subset of utterances considered important (summary-worthy) by the human judge. They are highlighted as black nodes on the grey line in Figure 2.2.

ABSTRACTIVE SUMMARY: A set of human-written abstractive sentences, represented as the nodes on the right in Figure 2.2. The abstractive summary itself consists of 4 sections:

- `abstract` (nodes in black): coherent text to summarize the meeting as a whole so that the content can be understood by those not present for the meeting.
- `decisions` (nodes in orange): all task-oriented decisions that were made during the meeting.
- `problems` (nodes in red): problems or difficulties encountered during the meeting, which came to the surface and remained open.
- `actions` (nodes in green): next steps that each member of the group will take before the next meeting.

ABSTRACTIVE-EXTRACTIVE LINKING: As shown in Figure 2.2, the utterances of the extractive summary are linked to the sentences of the abstractive summary according to whether the former can be jointly summarized by the latter. In other words, the former convey or support the information in the latter. A single utterance can be linked with one or more abstractive sentences, and vice versa (many-to-many mapping). All utterances linked to the same abstractive sentence form a single *abstractive community*.

By taking a close look at Figure 2.2, we can observe that for this specific example meeting: 1) `abstract` sentences summarize the entire content, since linked utterances are scattered across the entire meeting transcription. 2) `actions` sentences are linked to utterances located only at the end of the meeting, which is easy to understand since they correspond to assigned tasks to be performed before the next meeting. 3) `decisions` and `problems` sentences are linked to utterances in the second half of the meeting.

3

UNSUPERVISED ABSTRACTIVE MEETING SUMMARIZATION

In this chapter, we study the task of Abstractive Meeting Summarization, which aims to take a meeting transcription as input and generate an abstractive summary consisting of novel sentences as output. We introduce a graph-based framework for this task that is fully unsupervised and does not rely on any annotations. Our work combines the strengths of multiple recent approaches while addressing their weaknesses. Moreover, we leverage recent advances in word embeddings and graph degeneracy applied to NLP, to take exterior semantic knowledge into account and to design custom diversity and informativeness measures. Experiments on the AMI and ICSI corpus show that our system improves on the state-of-the-art. Code and data are publicly available [1], and our system can be interactively tested [2].

3.1 INTRODUCTION

People spend a lot of their time in meetings. The ubiquity of web-based meeting tools and the rapid improvement and adoption of Automatic Speech Recognition (ASR) is creating pressing needs for effective meeting speech summarization mechanisms.

Spontaneous multi-party meeting speech transcriptions widely differ from traditional documents. Instead of grammatical, well-segmented *sentences*, the input is made of often ill-formed and ungrammatical text fragments called *utterances*. On top of that, ASR transcription and segmentation errors inject additional noise into the input.

In this work, we combine the strengths of 6 approaches that had previously been applied to 3 different tasks (keyword extraction, multi-sentence compression, and summarization) into a unified, fully unsupervised meeting speech summarization framework that can generate readable summaries despite the noise inherent to ASR transcriptions. We also introduce some novel components. Our method reaches state-of-the-art performance and can be applied to languages other than English in an almost out-of-the-box fashion.

3.2 FRAMEWORK OVERVIEW

As illustrated in Figure 3.1, our system is made of 4 modules, briefly described in what follows.

1. https://bitbucket.org/dascim/acl2018_abssumm
2. http://datascience.open-paas.org/abs_summ_app

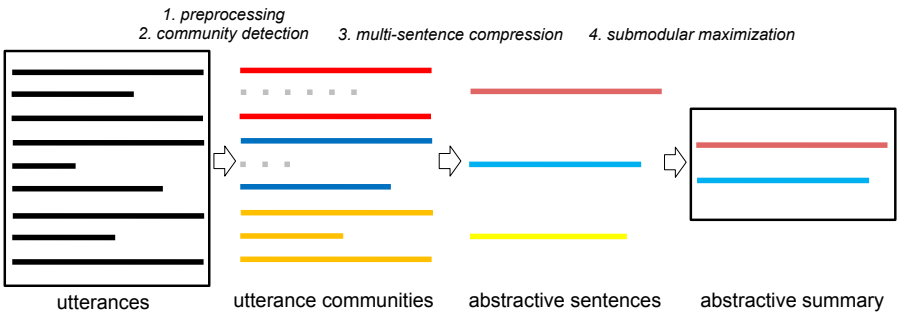

Figure 3.1 – Overarching system pipeline.

The first module pre-processes text. The goal of the second *Community Detection* step is to group together the utterances that should be summarized by a common abstractive sentence (Murray, Carenini, and Ng, 2012). These utterances typically correspond to a topic or subtopic discussed during the meeting. A single abstractive sentence is then separately generated for each community, using an extension of the Multi-Sentence Compression Graph (MSCG) of Filippova (2010). Finally, we generate a summary by selecting the best elements from the set of abstractive sentences under a budget constraint. We cast this problem as the maximization of a custom submodular quality function.

Note that our approach is fully unsupervised and does not rely on any annotations. Our input simply consists in a list of utterances without any metadata. All we need in addition to that is a part-of-speech tagger, a language model, a set of pre-trained word vectors, a list of stopwords and fillerwords, and optionally, access to a lexical database such as WordNet. Our system can work out-of-the-box in most languages for which such resources are available.

3.3 RELATED WORK AND CONTRIBUTIONS

As detailed below, our framework combines the strengths of 6 recent works. It also includes novel components.

Multi-Sentence Compression Graph (MSCG) (Filippova, 2010)

DESCRIPTION: a fully unsupervised, simple approach for generating a short, self-sufficient sentence from a cluster of related, overlapping sentences. As shown in Figure 3.5, a word graph is constructed with special edge weights, the K-shortest weighted paths are then found and re-ranked with a scoring function, and the best path is used as the compression. The assumption is that redundancy alone is enough to ensure informativeness and grammaticality.

LIMITATIONS: despite making great strides and showing promising results, Filippova (2010) reported that 48% and 36% of the generated sentences were missing important information and were not perfectly grammatical.

CONTRIBUTIONS: to respectively improve informativeness and grammaticality, we combine ideas found in Boudin and Morin (2013) and Mehdad et al. (2013), as described next.

More informative MSCG (Boudin and Morin, 2013)

DESCRIPTION: same task and approach as in Filippova (2010), except that a word co-occurrence network is built from the cluster of sentences, and that the PageRank scores of the nodes are computed in the manner of Mihalcea and Tarau (2004). The scores are then injected into the path re-ranking function to favor informative paths.

LIMITATIONS: PageRank is not state-of-the-art in capturing the importance of words in a document. Grammaticality is not considered.

CONTRIBUTIONS: we take grammaticality into account as in the work of Mehdad et al. (2013). We also follow recent evidence (Tixier, Malliaros, and Vazirgiannis, 2016) that *spreading influence*, as captured by graph degeneracy-based measures, is better correlated with "keywordedness" than PageRank scores, as explained next.

Graph-based word importance scoring (Tixier, Malliaros, and Vazirgiannis, 2016)

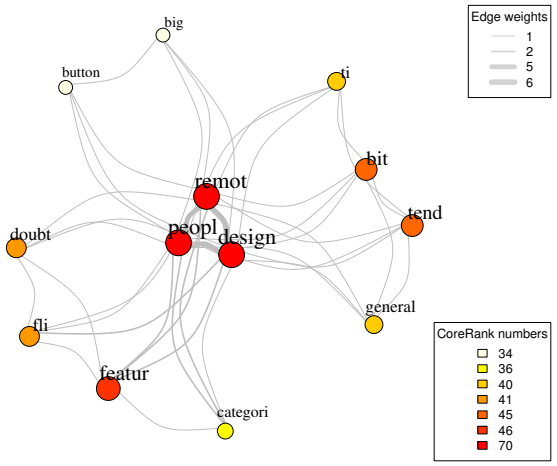

Figure 3.2 – Word co-occurrence network example, for the input text shown in Figure 3.5.

Word co-occurrence network (graph-of-words). As shown in Figure 3.2, we consider a word co-occurrence network as an undirected, weighted graph constructed by sliding a fixed-size window over text, and where edge weights represent co-occurrence counts (Mihalcea and Tarau, 2004; Tixier, Skianis, and Vazirgiannis, 2016). More details about the word co-occurrence network can be found in Subsection 2.1.2.

Important words are influential nodes. In social networks, it was shown that *influential spreaders*, that is, those individuals that can reach the largest part of the network in a given number of steps, are better identified via their core numbers rather than via their PageRank scores or degrees (Kitsak et al., 2010). See Figure 3.3 for the intuition. Similarly, in NLP, Tixier, Malliaros, and Vazirgiannis (2016) have shown that keywords are better identified via their core numbers rather than via their TextRank scores, that is, keywords are *influencers* within their word co-occurrence network.

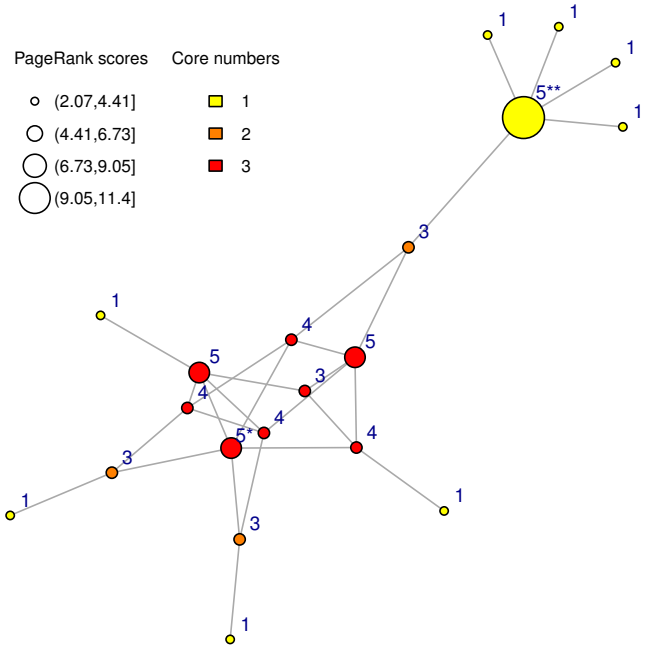

Figure 3.3 – k-core decomposition. The nodes $\star$ and $\star\star$ have same degree and similar PageRank numbers. However, node $\star$ is a much more influential spreader as it is strategically placed in the core of the network, as captured by its higher core number.

Graph degeneracy (Seidman, 1983). Let $G(V, E)$ be an undirected, weighted graph with $n = |V|$ nodes and $m = |E|$ edges. A k-core of G is a maximal subgraph of G in which every vertex v has at least weighted degree k. As shown in Figures 3.3 and 3.4, the k-core decomposition of G forms a hierarchy of nested subgraphs whose cohesiveness and size respectively increase and decrease with k. The higher-level cores can be viewed as a *filtered version* of the graph that excludes

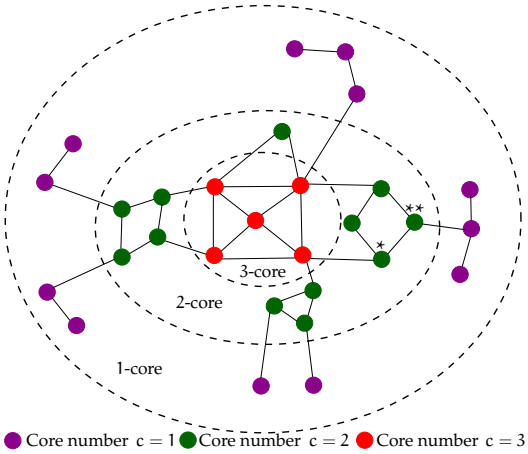

Figure 3.4 – Value added by CoreRank: while nodes ⋆ and ⋆⋆ have the same core number (=2), node ⋆ has a greater CoreRank score (3+2+2=7 vs 2+2+1=5), which better reflects its more central position in the graph.

noise. This property is highly valuable when dealing with graphs constructed from noisy text, like utterances. The core number of a node is the highest order of a core that contains this node.

The CoreRank number of a node (Bae and Kim, 2014; Tixier, Malliaros, and Vazirgiannis, 2016) is defined as the sum of the core numbers of its neighbors. As shown in Figure 3.4, CoreRank more finely captures the structural position of each node in the graph than raw core numbers. Also, stabilizing scores across node neighborhoods enhances the inherent noise robustness property of graph degeneracy, which is desirable when working with noisy speech-to-text output.

Time complexity. Building a graph-of-words is $O(nW)$, and computing the weighted k-core decomposition of a graph requires $O(m \log(n))$ (Batagelj and Zaveršnik, 2002). For small pieces of text, this two step process is so affordable that it can be used in real-time (Meladianos et al., 2017). Finally, computing CoreRank scores can be done with only a small overhead of $O(n)$, provided that the graph is stored as a hash of adjacency lists. Getting the CoreRank numbers from scratch for a community of utterances is therefore very fast, especially since typically in this context, $n \sim 10$ and $m \sim 100$.

Fluency-aware, more abstractive MSCG (Mehdad et al., 2013)

DESCRIPTION: a *supervised* framework for abstractive meeting summarization. Community detection is performed by (1) building an utterance graph with a logistic regression classifier, and (2) applying the CONGA algorithm. Then, before performing sentence compression with the MSCG, the authors also (3) build an entailment graph with a SVM classifier in order to eliminate redundant and less informative ut-

terances. In addition, the authors propose the use of WordNet (Miller, 1995) during the MSCG building phase to capture lexical knowledge between words and thus generate more abstractive compressions, and of a language model when re-ranking the shortest paths, to favor fluent compressions.

LIMITATIONS: this effort was a significant advance, as it was the first application of the MSCG to the meeting summarization task, to the best of our knowledge. However, steps (1) and (3) above are complex, based on handcrafted features, and respectively require annotated training data in the form of links between human-written abstractive sentences and original utterances and multiple external datasets (e.g., from the Recognizing Textual Entailment Challenge). Such annotations are costly to obtain and very seldom available in practice.

CONTRIBUTIONS: while we retain the use of WordNet and of a language model, we show that, without deteriorating the quality of the results, steps (1) and (2) above (community detection) can be performed in a much more simple, completely unsupervised way, and that step (3) can be removed. That is, the MSCG is powerful enough to remove redundancy and ensure informativeness, should proper edge weights and path re-ranking function be used.

In addition to the aforementioned contributions, we also introduce the following novel components into our abstractive summarization pipeline:

— we inject global exterior knowledge into the edge weights of the MSCG, by using the *Word Attraction Force* of Wang, Liu, and McDonald (2014), based on distance in the word embedding space,

— we add a diversity term to the path re-ranking function, that measures how many unique clusters in the embedding space are visited by each path,

— rather than using all the abstractive sentences as the final summary like in Mehdad et al. (2013), we maximize a custom submodular function to select a subset of abstractive sentences that is near-optimal given a budget constraint (summary size). A brief background of submodularity in the context of summarization is provided next.

Submodularity for summarization (Lin and Bilmes, 2010; Lin, 2012)

Selecting an optimal subset of abstractive sentences from a larger set can be framed as a budgeted submodular maximization task:

$$\underset{S \subseteq \mathcal{S}}{\operatorname{argmax}} f(S) | \sum_{s \in S} c_s \leq \mathcal{B} \tag{3.1}$$

where S is a summary, c_s is the cost (word count) of sentence s, $\mathcal{B}$ is the desired summary size in words (budget), and f is a summary quality scoring set function, which assigns a single numeric score to a summary S.

This combinatorial optimization task is NP-hard. However, near-optimal performance can be guaranteed with a modified greedy algorithm (Lin and Bilmes, 2010) that iteratively selects the sentence s that maximizes the ratio of quality function gain to scaled cost $f(S \cup s) - f(S) / c_s^r$ (where S is the current summary and $r \geq 0$ is a scaling factor).

In order for the performance guarantees to hold however, f has to be *submodular* and *monotone non-decreasing*. Our proposed f is described in Subsection 3.4.4.

3.4 OUR FRAMEWORK

We detail next each of the four modules in our architecture (shown in Figure 3.1).

3.4.1 *Text preprocessing*

We adopt preprocessing steps tailored to the characteristics of ASR transcriptions. Initial ellipsis, such as *'kay, 'til, and 'em*, is replaced receptively by its complete form *okay, until, and them*. Acronyms, such as *T_V_, V_C_R_, L_C_D_*, underlines are removed to make them compact. Consecutive repeated unigrams and bigrams are reduced to single terms. Specific ASR tags, such as {*vocalsound*}, {*pause*}, *and* {*gap*} are filtered out. In addition, filler words, such as *uh-huh*, *okay*, *well*, and *by the way* are also discarded. Consecutive stopwords at the beginning and end of utterances are stripped. In the end, utterances that contain less than 3 non-stopwords are pruned out. The surviving utterances are used for the next steps.

3.4.2 *Utterance community detection*

The goal here is to cluster utterances into communities that should be summarized by a common abstractive sentence.

We initially experimented with techniques capitalizing on word vectors (see Subsection 2.1.3 for more details), such as k-means and hierarchical clustering based on the Euclidean distance or the Word Mover's Distance (Kusner et al., 2015). We also tried graph-based approaches, such as community detection in a complete graph where nodes are utterances and edges are weighted based on the aforementioned distances.

Best results were obtained, however, with a simple approach in which utterances are projected into the vector space and assigned standard

TF-IDF weights (see Subsection 2.1.2 for more details). Then, the dimensionality of the utterance-term matrix is reduced with Latent Semantic Analysis (LSA), and finally, the k-means algorithm is applied. Note that LSA is only used here, during the utterance community detection phase, to remove noise and stabilize clustering. We do not use a topic graph in our approach.

We think using word embeddings was not effective, because in meeting speech, as opposed to traditional documents, participants tend to use the same term to refer to the same thing throughout the entire conversation, as noted by Riedhammer, Favre, and Hakkani-Tür (2010), and as verified in practice. This is probably why, for clustering utterances, capturing synonymy is counterproductive, as it artificially reduces the distance between every pair of utterances and blurs the picture.

3.4.3 *Multi-sentence compression*

The following steps are performed separately for each community.

Word importance scoring

From a processed version of the community (stemming and stopword removal), we construct an undirected, weighted word co-occurrence network as described in Section 3.3. We use a sliding window of size $W = 6$ not overspanning utterances. Note that stemming is performed only here, and for the sole purpose of building the word co-occurrence network.

We then compute the CoreRank numbers of the nodes as described in Section 3.3.

We finally reweigh the CoreRank scores, indicative of word importance within a given community, with a quantity akin to an *Inverse Document Frequency*, where communities serve as documents and the full meeting as the collection. We thus obtain something equivalent to the TW-IDF weighting scheme of Rousseau and Vazirgiannis (2013), where the CoreRank scores are the term weights TW:

$$TW\text{-}IDF(t, d, D) = TW(t, d) \times IDF(t, D) \tag{3.2}$$

where t is a term belonging to community d, and D is the set of all utterance communities. We compute the IDF as $IDF(t, D) = 1 + \log^{|D|}/_{D_t}$, where $|D|$ is the number of communities and D_t the number of communities containing t.

The intuition behind this reweighing scheme is that a term should be considered important within a given meeting if it has a high CoreRank score within its community *and* if the number of communities in which the term appears is relatively small. A detailed explanation of TW-IDF is provided in Subsection 2.1.2.

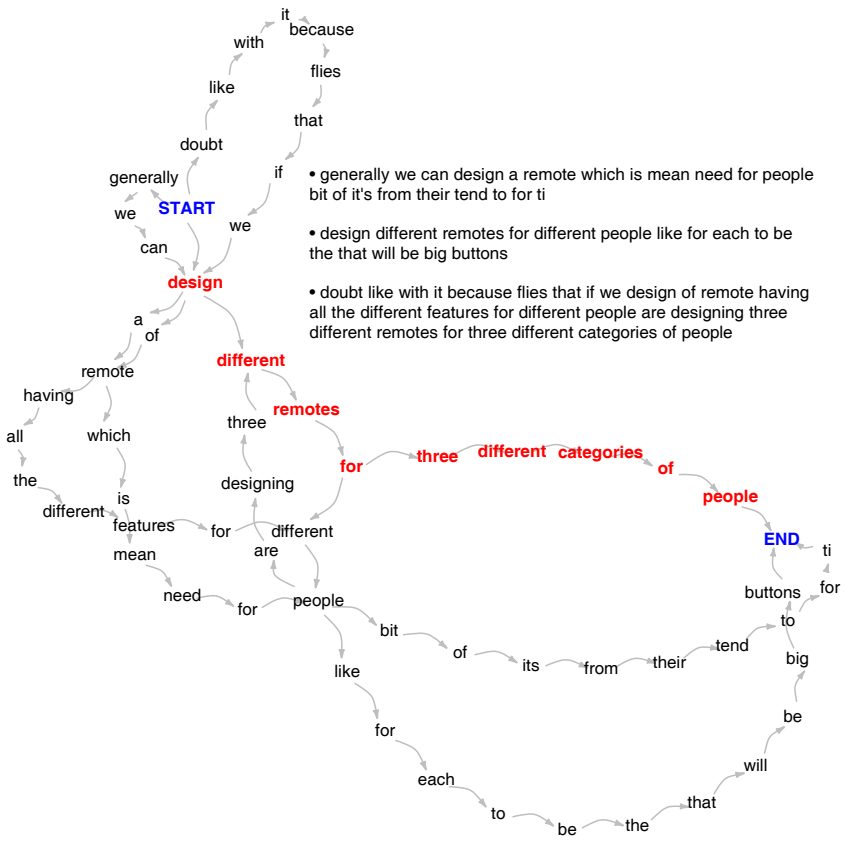

Figure 3.5 – Compressed sentence (in **bold red**) generated by our multi-sentence compression graph (MSCG) for a 3-utterance community from meeting IS1009b of the AMI corpus. Using Filippova (2010)'s weighting and re-ranking scheme here would have selected another path: *design different remotes for different people bit of it's from their tend to for ti*. Note that the compressed sentence does not appear in the initial set of utterances, and is compact and grammatical, despite the redundancy, transcription and segmentation errors of the input. The *abstractive* and *robust* nature of the MSCG makes it particularly well-suited to the meeting domain.

Word graph building

The backbone of the graph is laid out as a directed sequence of nodes corresponding to the words in the first utterance, with special `START` and `END` nodes at the beginning and at the end (see Figure 3.5). Edge direction follows the natural flow of text. Words from the remaining utterances are then iteratively added to the graph (between the `START` and `END` nodes) based on the following rules:

1) if the word is a **non-stopword**, the word is mapped onto an existing node if it has the same lowercased form and the same part-of-

speech tag [3]. In case of multiple matches, we check the immediate context (the preceding and following words in the utterance and the neighboring nodes in the graph), and we pick the node with the largest context overlap or which has the greatest number of words already mapped to it (when no overlap). When there is no match, we use WordNet as described below.

2) if the word is a **stopword** and there is a match, it is mapped only if there is an overlap of at least one non-stopword in the immediate context. Otherwise, a new node is created.

Finally, note that any two words appearing within the same utterance cannot be mapped to the same node. This ensures that every utterance is a loopless path in the graph. Of course, there are many more paths in the graphs than original utterances.

Use of WordNet

When the word to be mapped to the MSCG is a **non-stopword**, and if there is no node in the graph that has the same lowercased form and the same part-of-speech tag, we try to perform the mapping by using WordNet in the following order:

(i) there is a node which is a synonym of the word (e.g., "price" and "costs"). The word is mapped to that node, and the node is relabeled with the word if the latter has a higher TW-IDF score.

(ii) there is a node which is a hypernym of the word (e.g., "diamond" and "gemstone"). The word is mapped to that node, and the node is relabeled with the word if the latter has a higher TW-IDF score.

(iii) there is a node which shares a common hypernym with the word (e.g., "red","blue" → "color"). If the product of the WordNet path distance similarities of the common hypernym with the node and the word exceeds a certain threshold, the word is mapped to that node and the node is relabeled with the hypernym. A completely new word might thus be introduced. We set its TW-IDF score as the highest TW-IDF of the two words it replaces. When multiple nodes are eligible for mapping, we select the one with greatest path distance similarity product.

(iv) there is a node which is in an entailment relation with the word (e.g., "look" is entailed by "see"). The word is mapped to that node, and the node is relabeled with the word if the latter has a higher TW-IDF score.

In attempts **i**, **ii**, and **iv** above, if there is more than one candidate node, we select the one with highest TW-IDF score. If all attempts above are unsuccessful, a new node is created for the word.

3. We used NLTK's averaged perceptron tagger, available at: http://www.nltk.org/api/nltk.tag.html#module-nltk.tag.perceptron

Edge weight assignment

Once the word graph is constructed, we assign weights to its edges as:

$$w'''(p_i, p_j) = \frac{w'(p_i, p_j)}{w''(p_i, p_j)} \tag{3.3}$$

where p_i and p_j are two neighbors in the MSCG. As detailed next, those weights combine *local co-occurrence statistics* (numerator) with *global exterior knowledge* (denominator). Note that the lower the weight of an edge, the better.

Local co-occurrence statistics. We use Filippova (2010)'s formula:

$$w'(p_i, p_j) = \frac{f(p_i) + f(p_j)}{\sum_{P \in G', p_i, p_j \in P} \text{diff}(P, p_i, p_j)^{-1}} \tag{3.4}$$

where $f(p_i)$ is the number of words mapped to node p_i in the MSCG G', and $\text{diff}(P, p_i, p_j)^{-1}$ is the inverse of the distance between p_i and p_j in a path P (in number of hops). This weighting function favors edges between infrequent words that frequently appear close to each other in the text (the lower, the better).

Global exterior knowledge. We introduce a second term based on the *Word Attraction Force score* of Wang, Liu, and McDonald (2014):

$$w''(p_i, p_j) = \frac{f(p_i) \times f(p_j)}{d^2_{p_i, p_j}} \tag{3.5}$$

where d_{p_i, p_j} is the Euclidean distance between the words mapped to p_i and p_j in a word embedding space [4]. This component favors paths going through salient words that have *high semantic similarity* (the higher, the better). The goal is to ensure readability of the compression, by avoiding to generate a sentence jumping from one word to a completely unrelated one.

Path re-ranking

As in Boudin and Morin (2013), we use a shortest weighted path algorithm to find the K paths between the `START` and `END` symbols having the lowest cumulative edge weight:

$$W(P) = \sum_{i=1}^{|P|-1} w'''(p_i, p_{i+1}) \tag{3.6}$$

4. GoogleNews vectors https://code.google.com/archive/p/word2vec

Where $|P|$ is the number of nodes in the path. Paths having less than z words or that do not contain a verb are filtered out (z is a tuning parameter). However, unlike in Boudin and Morin (2013), we rerank the K best paths with the following novel weighting scheme (the lower, the better), and the path with the lowest score is used as the compression:

$$\text{score}(P) = \frac{W(P)}{|P| \times F(P) \times C(P) \times D(P)} \tag{3.7}$$

The denominator takes into account the length of the path, and its fluency (F), coverage (C), and diversity (D). F, C, and D are detailed in what follows.

Fluency. We estimate the grammaticality of a path with an n-gram language model. In our experiments, we used a trigram model [5]:

$$F(P) = \frac{\sum_{i=1}^{|P|} \log Pr(p_i | p_{i-n+1}^{i-1})}{\#n\text{-}gram} \tag{3.8}$$

where $|P|$ denote path length, and p_i and $\#n\text{-}gram$ are respectively the words and number of n-grams in the path.

Coverage. We reward the paths that visit important nouns, verbs and adjectives:

$$C(P) = \frac{\sum_{p_i \in P} \text{TW-IDF}(p_i)}{\#p_i} \tag{3.9}$$

where $\#p_i$ is the number of nouns, verbs and adjectives in the path. The TW-IDF scores are computed as explained in Subsection 3.4.3.

Diversity. We cluster all words from the MSCG in the word embedding space by applying the k-means algorithm. We then measure the diversity of the vocabulary contained in a path as the number of unique clusters visited by the path, normalized by the length of the path:

$$D(P) = \frac{\sum_{j=1}^{k} 1_{\exists p_i \in P | p_i \in \text{cluster}_j}}{|P|} \tag{3.10}$$

The graphical intuition for this measure is provided in Figure 3.6. Note that we do not normalize D by the total number of clusters (only by path length) because k is fixed for all candidate paths.

3.4.4 *Budgeted submodular maximization*

We apply the previous steps separately for all utterance communities, which results in a set $\mathcal{S}$ of abstractive sentences (one for each community). This set of sentences can already be considered to be a summary of the meeting. However, it might exceed the maximum

5. CMUSphinx English LM: https://cmusphinx.github.io

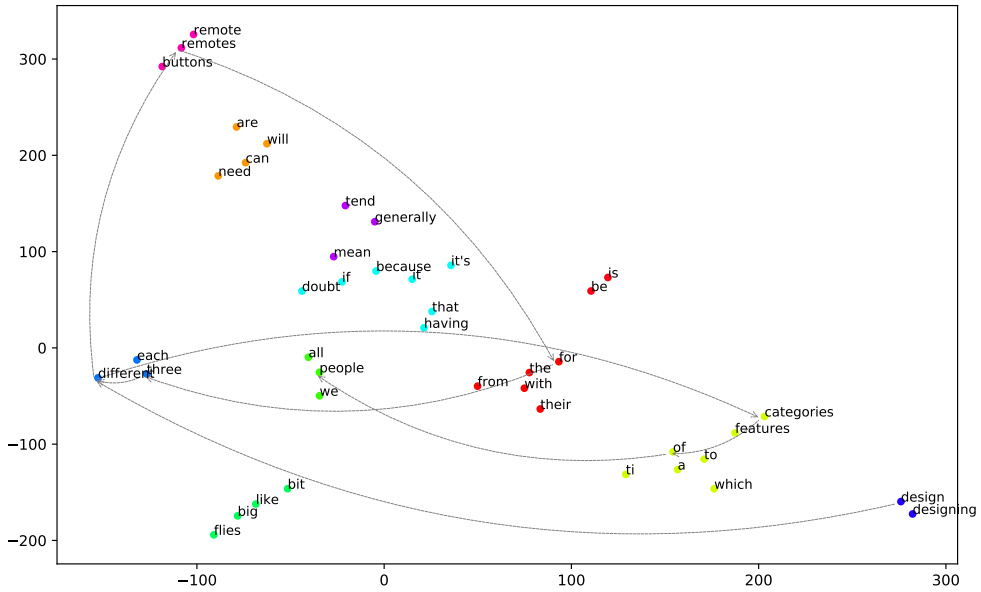

Figure 3.6 – t-SNE visualization (Maaten and Hinton, 2008) of the Google News vectors of the words in the utterance community shown in Figure 3.5. Arrows join the words in the best compression path shown in Figure 3.5. Movements in the embedding space, as measured by the number of unique clusters covered by the path (here, 6/11), provide a sense of the diversity of the compressed sentence, as formalized in Equation 3.10.

size allowed, and still contain some redundancy or off-topic sections unrelated to the general theme of the meeting (e.g., chit-chat).

Therefore, we design the following *submodular* and *monotone non-decreasing* objective function:

$$f(S) = \sum_{s_i \in S} n_{s_i} w_{s_i} + \lambda \sum_{j=1}^{k} 1_{\exists s_i \in S | s_i \in cluster_j} \tag{3.11}$$

where $\lambda \geq 0$ is the trade-off parameter, n_{s_i} is the number of occurrences of word s_i in S, and w_{s_i} is the CoreRank score of s_i.

Then, as explained in Section 3.3, we obtain a near-optimal subset of abstractive sentences by maximizing f with a greedy algorithm. CoreRank scores and clusters are found as previously described, except that this time they are obtained from the full processed meeting transcription rather than from a single utterance community.

3.5 EXPERIMENTAL SETUP

3.5.1 *Datasets*

We conducted experiments on the widely-used AMI (McCowan et al., 2005) and ICSI (Janin et al., 2003) benchmark datasets (see Subsection 2.3 for more information). We used the traditional test sets of

20 and 6 meetings respectively for the AMI and ICSI corpora (Riedhammer et al., 2008). Each meeting in the AMI test set is associated with a human abstractive summary of 290 words on average, whereas each meeting in the ICSI test set is associated with 3 human abstractive summaries of respective average sizes 220, 220 and 670 words.

For parameter tuning, we constructed development sets of 47 and 25 meetings, respectively for AMI and ICSI, by randomly sampling from the training sets. The word error rate of the ASR transcriptions is respectively of 36% and 37% for AMI and ICSI.

3.5.2 *Baselines*

We compared our system against 7 baselines listed below. Note that preprocessing was exactly the same for our system and all baselines.

— **Random**. A basic baseline recommended by Riedhammer et al. (2008) to ease cross-study comparison. This system randomly selects utterances without replacement from the transcription until the budget is violated. To account for stochasticity, we report scores averaged over 30 runs.

— **Longest Greedy**. A basic baseline recommended by Riedhammer et al. (2008) to ease cross-study comparison. The longest remaining utterance is selected at each step from the transcription until the summary size constraint is satisfied.

— **TextRank** (Mihalcea and Tarau, 2004). Utterances within the transcription are represented as nodes in an undirected complete graph, and edge weights are assigned based on lexical similarity between utterances. To provide a summary, the top nodes according to the weighted PageRank algorithm (Page et al., 1999) are selected. We used a publicly available implementation [6].

— **ClusterRank** (Garg et al., 2009). This system is an extension of TextRank to meeting summarization. Firstly, utterances are segmented into clusters. A complete graph is built from the clusters. Then, a score is assigned to each utterance based on both the PageRank score of the cluster it belongs to and its cosine similarity with the cluster centroid. In the end, a greedy selection strategy is applied to build the summary out of the highest scoring utterances. Since the authors did not make their code publicly available and were not able to share it privately, we wrote our own implementation.

— **CoreRank submodular & PageRank submodular** (Tixier, Meladianos, and Vazirgiannis, 2017). These two *extractive* baselines implement the last step of our pipeline (see Subsection 3.4.4). That is, budgeted submodular maximization is applied directly on the full list of utterances. As can be inferred from their names, the only dif-

6. https://github.com/summanlp/textrank

ference between those two baselines is that the first uses PageRank scores, whereas the second uses CoreRank scores.

— **Oracle**. This system is the same as the Random baseline, but instead of sampling utterances from the ASR transcription, it draws from the human extractive summaries. Annotators put those summaries together by selecting the best utterances from the entire manual transcription. Scores were averaged over 30 runs due to the randomness of the procedure.

In addition to the baselines above, we included in our comparison 3 variants of our system using different MSCGs: **Our System (Baseline)** uses the original MSCG of Filippova (2010), **Our System (KeyRank)** uses that of Boudin and Morin (2013), and **Our System (FluCovRank)** that of Mehdad et al. (2013). Details about each approach were given in Section 3.3.

3.5.3 *Parameter tuning*

For *Our System* and each of its variants, we conducted a grid search on the development sets of each corpus, for fixed summary sizes of 350 and 450 words (AMI and ICSI). We searched the following parameters:

— n: number of utterance communities (see Subsection 3.4.2). We tested values of n ranging from 20 to 60, with steps of 5. This parameter controls how much abstractive should the summary be. If all utterances are assigned to their own singleton community, the MSCG is of no utility, and our framework is extractive. It becomes more and more abstractive as the number of communities decreases.

— z: minimum path length (see Subsection 3.4.3). We searched values in the range $[6, 16]$ with steps of 2. If a path is shorter than a certain minimum number of words, it often corresponds to an invalid sentence, and should thereby be filtered out.

— λ and r, the trade-off parameter and the scaling factor (see Subsection 3.4.4). We searched $[0, 1]$ and $[0, 2]$ (respectively) with steps of 0.1. The parameter λ plays a regularization role favoring diversity. The scaling factor makes sure the quality function gain and utterance cost are comparable.

The best parameter values for each corpus are summarized in Table 3.1. λ is mostly non-zero, indicating that it is necessary to include a regularization term in the submodular function. In some cases though, r is equal to zero, which means that utterance costs are not involved in the greedy decision heuristic. These observations contradict the conclusion of Lin (2012) that $r = 0$ cannot give best results.

Apart from the tuning parameters, we set the number of LSA dimensions to 30 and 60 (resp. on AMI and ISCI). The small number of LSA

System	AMI	ICSI
Our System	50, 8, (0.7, 0.5)	40, 14, (0.0, 0.0)
Our System (Baseline)	50, 12, (0.3, 0.5)	45, 14, (0.1, 0.0)
Our System (KeyRank)	50, 10, (0.2, 0.9)	45, 12, (0.3, 0.4)
Our System (FluCovRank)	35, 6, (0.4, 1.0)	50, 10, (0.2, 0.3)

Table 3.1 – Optimal parameter values $n, z, (\lambda, r)$.

dimensions retained can be explained by the fact that the AMI and ICSI transcriptions feature 532 and 1126 unique words on average, which is much smaller than traditional documents. This is due to relatively small meeting duration, and to the fact that participants tend to stick to the same terms throughout the entire conversation. For the k-means algorithm, k was set equal to the minimum path length z when doing MSCG path re-ranking (see Equation 3.10), and to 60 when generating the final summary (see Equation 3.11).

Following Boudin and Morin (2013), the number of shortest weighted paths K was set to 200, which is greater than the $K = 100$ used by Filippova (2010). Increasing K from 100 improves performance with diminishing returns, but significantly increases complexity. We empirically found 200 to be a good trade-off.

3.6 RESULTS AND INTERPRETATION

Metrics. We evaluated performance with the widely-used ROUGE-1, ROUGE-2 and ROUGE-SU4 metrics (Lin, 2004). These metrics are respectively based on unigram, bigram, and unigram plus skip-bigram overlap with maximum skip distance of 4, and have been shown to be highly correlated with human evaluations (Lin, 2004). ROUGE-2 scores can be seen as a measure of summary readability (Lin and Hovy, 2003; Ganesan, Zhai, and Han, 2010). ROUGE-SU4 does not require consecutive matches but is still sensitive to word order. Further explanation of ROUGE is available in Subsection 2.2.2.

Macro-averaged results for summaries generated from *automatic transcriptions* can be seen in Figure 3.7 and Table 3.2. Table 3.2 provides detailed comparisons over the fixed budgets that we used for parameter tuning, while Figure 3.7 shows the performance of the models for budgets ranging from 150 to 500 words.

ROUGE-1. Our systems outperform all baselines on AMI (including *Oracle*) and all baselines on ICSI (except *Oracle*). Specifically, *Our System* is best on ICSI, while *Our System* (*KeyRank*) is superior on AMI. We can also observe on Figure 3.7 that our systems are consistently better throughout the different summary sizes, even though their parameters were tuned for specific sizes only. This shows that the best parameter values are quite robust across the entire budget range.

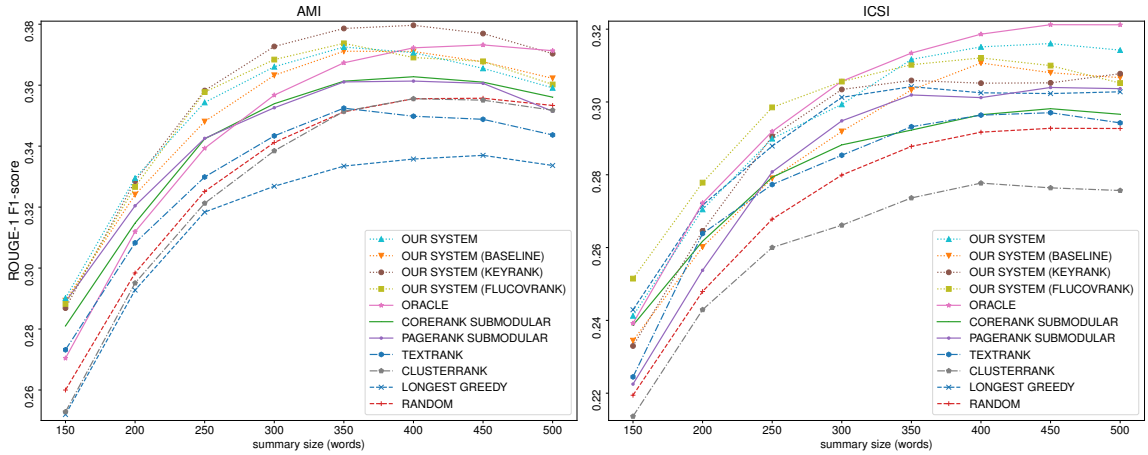

Figure 3.7 – ROUGE-1 F-1 scores for various budgets (ASR transcriptions).

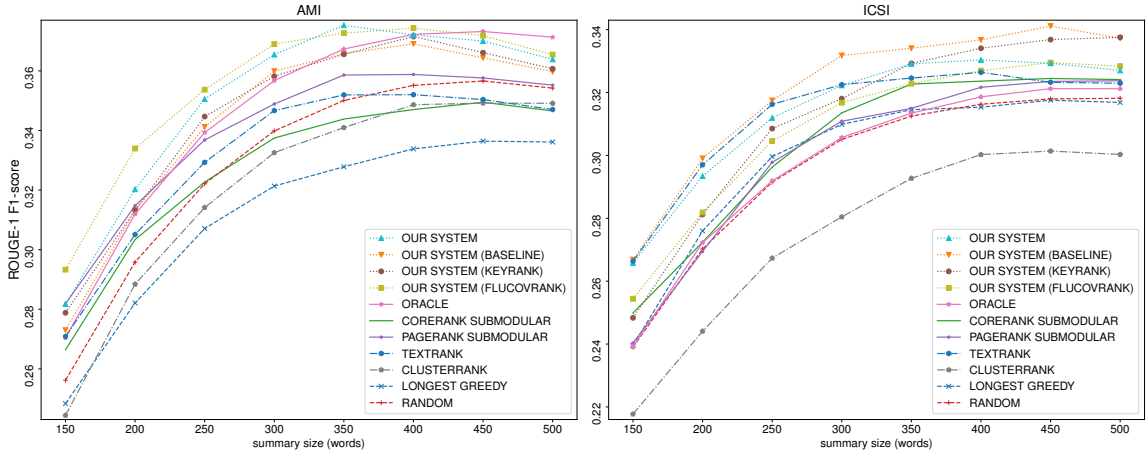

Figure 3.8 – ROUGE-1 F-1 scores for various budgets (manual transcriptions).

ROUGE-2. Again, our systems (except *Our System* (*Baseline*)) outperform all baselines, except *Oracle*. In addition, *Our System* and *Our System* (*FluCovRank*) consistently improve on *Our System* (*Baseline*), which proves that the novel components we introduce improve summary fluency.

ROUGE-SU4. ROUGE-SU4 was used to measure the amount of in-order word pairs overlapping. Our systems are competitive with all baselines, including *Oracle*. Like with ROUGE-1, *Our System* is better than *Our System* (*KeyRank*) on ICSI, whereas the opposite is true on AMI.

General remarks.

— The summaries of all systems except *Oracle* were generated from noisy ASR transcriptions, but were compared against human abstractive summaries. ROUGE being based on word overlap, it makes it very difficult to reach very high scores, because many

	AMI ROUGE-1			AMI ROUGE-2			AMI ROUGE-SU4			ICSI ROUGE-1			ICSI ROUGE-2			ICSI ROUGE-SU4		
	R	P	F-1	R	P	F-1	R	P	F-1	R	P	F-1	R	P	F-1	R	P	F-1
Our System	41.83	34.44	37.25	8.22	6.95	7.43	15.83	13.70	14.51	36.99	28.12	**31.60**	5.41	4.39	4.79	13.10	10.17	**11.35**
Our System (Baseline)	41.56	34.37	37.11	7.88	6.66	7.11	15.36	13.20	14.02	36.39	27.20	30.80	5.19	4.12	4.55	12.59	9.70	10.86
Our System (KeyRank)	42.43	35.01	**37.86**	8.72	7.29	**7.84**	16.19	13.76	**14.71**	35.95	27.00	30.52	4.64	3.64	4.04	12.43	9.23	10.50
Our System (FluCovRank)	41.84	34.61	37.37	8.29	6.92	7.45	16.28	13.48	14.58	36.27	27.56	31.00	5.56	4.35	**4.83**	13.47	9.85	11.29
Oracle	40.49	34.65	**36.73**	8.07	7.35	**7.55**	15.00	14.03	**14.26**	37.91	28.39	**32.12**	5.73	4.82	**5.18**	13.35	10.73	**11.80**
CoreRank Submodular	41.14	32.93	36.13	8.06	6.88	7.33	14.84	13.91	14.18	35.22	26.34	29.82	4.36	3.76	4.00	12.11	9.58	10.61
PageRank Submodular	40.84	33.08	36.10	8.27	6.88	7.42	15.37	13.71	14.32	36.05	26.69	30.40	4.82	4.16	4.42	12.19	10.39	11.14
TextRank	39.55	32.60	35.25	7.67	6.43	6.90	14.87	12.87	13.62	34.89	26.33	29.70	4.60	3.74	4.09	12.42	9.43	10.64
ClusterRank	39.36	32.53	35.14	7.14	6.05	6.46	14.34	12.80	13.35	32.63	24.44	27.64	4.03	3.44	3.68	11.04	8.88	9.77
Longest Greedy	37.31	30.93	33.35	5.77	4.71	5.11	13.79	11.11	12.15	35.57	26.74	30.23	4.84	3.88	4.27	13.09	9.46	10.90
Random	39.42	32.48	35.13	6.88	5.89	6.26	14.07	12.70	13.17	34.78	25.75	29.28	4.19	3.51	3.78	11.61	9.37	10.29

Table 3.2 – Macro-averaged results for 350 and 450 word summaries (ASR transcriptions).

	AMI ROUGE-1			AMI ROUGE-2			AMI ROUGE-SU4			ICSI ROUGE-1			ICSI ROUGE-2			ICSI ROUGE-SU4		
	R	P	F-1	R	P	F-1	R	P	F-1	R	P	F-1	R	P	F-1	R	P	F-1
Our System	42.03	34.77	**37.53**	8.87	7.56	8.06	15.92	14.08	**14.76**	38.57	29.30	32.93	5.80	4.74	5.14	13.92	10.79	12.04
Our System (Baseline)	40.88	33.96	36.58	8.13	6.95	7.39	15.17	13.25	13.97	40.03	30.20	**34.11**	6.65	5.51	**5.98**	14.65	11.37	**12.70**
Our System (KeyRank)	40.87	33.91	36.56	8.42	7.12	7.62	15.50	13.48	14.25	39.55	29.79	33.68	6.32	5.19	5.64	14.63	10.99	12.47
Our System (FluCovRank)	41.73	34.50	37.27	8.45	7.05	7.60	16.08	13.47	14.49	38.57	29.21	32.95	6.38	5.08	5.60	14.38	10.62	12.13
Oracle	40.49	34.65	**36.73**	8.07	7.35	**7.55**	15.00	14.03	**14.26**	37.91	28.39	**32.12**	5.73	4.82	**5.18**	13.35	10.73	**11.80**
CoreRank Submodular	38.95	31.49	34.38	7.85	6.81	7.20	14.08	13.55	13.61	37.31	29.51	32.45	5.59	5.05	5.24	13.19	11.08	11.87
PageRank Submodular	40.58	32.87	35.86	9.20	7.77	**8.32**	15.59	14.14	14.64	37.72	28.86	32.35	6.35	5.46	5.82	13.35	11.60	12.30
TextRank	39.47	32.57	35.19	7.74	6.62	7.05	14.80	13.03	13.69	37.60	28.79	32.32	6.63	5.53	**5.98**	14.18	11.18	12.41
ClusterRank	38.32	31.51	34.10	6.93	5.95	6.31	13.69	12.40	12.84	35.66	26.58	30.14	4.53	3.99	4.21	12.10	9.71	10.69
Longest Greedy	36.73	30.39	32.78	5.52	4.58	4.93	13.52	10.91	11.93	37.15	28.21	31.76	5.50	4.60	4.98	13.59	10.03	11.46
Random	39.29	32.38	35.01	7.14	6.16	6.52	14.16	12.95	13.35	37.48	28.10	31.80	5.41	4.65	4.95	12.97	10.67	11.61

Table 3.3 – Macro-averaged results for 350 and 450 word summaries (manual transcriptions).

words in the ground truth summaries do not appear in the transcriptions at all.

— The scores of all systems are lower on ICSI than on AMI. This can be explained by the fact that on ICSI, the system summaries have to jointly match (at the time of parameter tuning) 3 human abstractive summaries of different content and size, which is much more difficult than matching a single summary.

— Our framework is very competitive to *Oracle*, which is notable since the latter has direct access to the human extractive summaries. Note that *Oracle* does not reach very high ROUGE scores because the overlap between the human extractive and abstractive summaries is low (19% and 29%, respectively on AMI and ICSI test sets).

In addition to the above results obtained on automatic transcriptions, the same information for summaries generated from *manual transcriptions* is provided in Figure 3.8 and Table 3.3. Finally, summary examples are available in Section 3.8.

3.7 CONCLUSION AND FUTURE WORK

Our framework combines the strengths of 6 approaches that had previously been applied to 3 different tasks (keyword extraction, multi-sentence compression, and summarization) into a unified, fully unsupervised summarization framework, and introduces some novel components. Rigorous evaluation on the AMI and ICSI corpora shows that we reach state-of-the-art performance, and generate reasonably grammatical abstractive summaries despite taking noisy utterances as input and not relying on any annotations or training data. Finally, thanks to its fully unsupervised nature, our method is applicable to other languages than English in an almost out-of-the-box manner.

Our framework was developed for the meeting domain. Indeed, our generative component, the multi-sentence compression graph (MSCG), needs redundancy to perform well. Such redundancy is typically present in meeting speech but not in traditional documents. In addition, the MSCG is by design robust to noise, and our custom path re-ranking strategy, based on graph degeneracy, makes it even more robust to noise. As a result, our framework is advantaged on ASR input. Finally, we use a language model to favor fluent paths, which is crucial when working with (meeting) speech but not that important when dealing with well-formed input.

Future efforts should be dedicated to improving the community detection phase (we introduce a novel approach to this task in Chapter 5) and generating more abstractive sentences, probably by harnessing deep learning. However, the lack of large training sets for the meeting domain is an obstacle to the use of neural approaches.

3.8 EXAMPLE SUMMARIES

Examples were generated from the manual transcriptions of the TS3003c AMI meeting. Note that our system can also be interactively tested at: http://datascience.open-paas.org/abs_summ_app.

Reference Summary (254 words)

The project manager opened the meeting and recapped the decisions made in the previous meeting.
The marketing expert discussed his personal preferences for the design of the remote and presented the results of trend-watching reports, which indicated that there is a need for products which are fancy, innovative, easy to use, in dark colors, in recognizable shapes, and in a familiar material like wood.
The user interface designer discussed the option to include speech recognition and which functions to include on the remote.
The industrial designer discussed which options he preferred for the remote in terms of energy sources, casing, case supplements, buttons, and chips.
The team then discussed and made decisions regarding energy sources, speech recognition, LCD screens, chips, case materials and colors, case shape and orientation, and button orientation.
The team members will look at the corporate website.
The user interface designer will continue with what he has been working on.
The industrial designer and user interface designer will work together.
The remote will have a docking station.
The remote will use a conventional battery and a docking station which recharges the battery.
The remote will use an advanced chip.
The remote will have changeable case covers.
The case covers will be available in wood or plastic.
The case will be single curved.
Whether to use kinetic energy or a conventional battery with a docking station which recharges the remote.
Whether to implement an LCD screen on the remote.
Choosing between an LCD screen or speech recognition.
Using wood for the case.

Our System (250 words)

attract elderly people can use the remote control
changing channels button on the right side that would certainly yield great options for the design of the remote
personally i dont think that older people like to shake your remote control
imagine that the remote control and the docking station
remote control have to lay in your hand and right hand users
finding an attractive way to control the remote control
casing the manufacturing department can deliver a flat casing single or double curved casing
top of that the lcd screen would help in making the remote control easier
increase the price for which were selling our remote control
remote controls are using a onoff button still on the top
apply remote control on which you can apply different case covers
button on your docking station which you can push and then it starts beeping

surveys have indicated that especially wood is the material for older people
mobile phones so like the nokia mobile phones when you can change the case
greyblack colour for people prefer dark colours
brings us to the discussion about our concepts
docking station and small screen would be our main points of interest
industrial designer and user interface designer are going to work
innovativeness was about half of half as important as the fancy design
efficient and cheaper to put it in the docking station
case supplement and the buttons it really depends on the designer
start by choosing a case
deployed some trendwatchers to milan

Our System (Baseline) (250 words)

apply remote controls on which you can apply different case for his remote control
changing channels and changing volume button on both sides that would certainly yield great options for the design of the remote
personally i dont think that older people like to shake their remote control
finding an attractive way to control the remote control the i found some something about speech recognition
imagine that the remote control and the docking station should be telephone-shaped
casing the manufacturing department can deliver a flat casing single or double curved casing
remote control have to lay in your hand and right hand users
remote controls are using a onoff button over in this corner
woodlike for the more exclusive people can use the remote control
heard our industrial designer talk about flat single curved and double curved
innovativeness this means functions which are not featured in other remote control
button on your docking station which you can push and then it starts beeping
greyblack colour for people prefer dark colours
docking station and small screen would be our main points of interest
special button for subtitles for people which c f who cant read small subtitles
pretty big influence on production price and image unless we would start two product lines
surveys have indicated that especially wood is the material for older people
mobile phones so like the nokia mobile phones when you can change the case
case the supplement and the buttons it really depends on the designer
buttons

Our System (KeyRank) (250 words)

changing case covers
prefer a design where the remote control and the docking station
greyblack colour for people prefer dark colours
remote controls are using a onoff button over in this corner
requirements are teletext docking station and small screen with some extras that button information
apply remote controls on which you can apply different case covers
woodlike for the more exclusive people can use the remote control
casing the manufacturing department can deliver a flat casing single or double curved casing

remote control have to lay in your hand and right hand users
asked if w they would if people would pay more for speech recognition function would not make the remote control
start by choosing a case
innovativeness this means functions which are not featured in other remote controls
top of that the lcd screen would help in making the remote control easier
changing channels and changing volume button on both sides that would certainly yield great options for the design of the remote
personally i dont think that older remotes are flat board smartboard
button on your docking station which you can push and then it starts beeping
case supplement and the buttons it really depends on the designer
surveys have indicated that especially wood is the material for older people will recognise the button
speak speech recognition and a special button for subtitles for people which c f who cant read small subtitles
innovativeness was about half as important as the fancy design
pretty big influence

Our System (FluCovRank) (250 words)

elderly people can use the remote control
remote controls are using a onoff button still on the top
general idea of the concepts and the material for older people like to shake your remote control
docking station and small screen would be our main points of interest
industrial designer and user interface designer are going to work
casing the manufacturing department can deliver single curved
changing channels and changing volume button on both side that would certainly yield great options for the design of the remote
button on your docking station which you can push and then it starts beeping
imagine that the remote control will be standing up straight in the docking station will help them give the remote
asked if w they would if people would pay more for speech recognition in a remote control you can call it and it gives an sig signal
research about bi large lcd sh display for for displaying the the functions of the buttons
case the supplement and the buttons it really depends on the designer
pointed out earlier that a lot of remotes rsi
innovativeness was about half of half as important as the fancy design
push on the button for subtitles for people which c f who cant read small subtitles
efficient and cheaper to put it in the docking station could be one of the marketing issues
difficult to handle and to get in the right shape to older people
talk about the energy source is rather fancy

4

DIALOGUE ACT CLASSIFICATION

In this chapter, we deal with the task of Dialogue Act (DA) classification, whose goal is to assign each utterance a label to represent its communicative intention. This task is considered as the first stepping stone for further discourse analysis and understanding. Recent work approaches the task as a sequence labeling problem, using neural network models coupled with a Conditional Random Field (CRF) as the last layer. CRF models the conditional probability of the target DA label sequence given the input utterance sequence. However, the task involves another important input sequence, that of speakers, which is ignored by previous work. To address this limitation, this work proposes a simple modification of the CRF layer that takes speaker-change into account. Experiments on the SwDA corpus show that our modified CRF layer outperforms the original one, with very wide margins for some DA labels. Further, visualizations demonstrate that our CRF layer can learn meaningful, sophisticated transition patterns between DA label pairs conditioned on speaker-change in an end-to-end way. Code is publicly available [1].

4.1 INTRODUCTION

A conversation can be seen as a sequence of utterances. The task of dialogue act classification aims at assigning to each utterance a DA label to represent its communicative intention. Dialogue acts originate from the notion of *illocutionary force* (speaker's intention in delivering an utterance) introduced back in the theory of Speech Act (Austin, 1962; Searle, 1969). DAs are assigned based on a combination of syntactic, semantic, and pragmatic criteria (Stolcke et al., 2000). As shown in Table 4.1, some examples of DAs include stating, questioning, answering, etc. The full set of DA labels is predefined. A number of annotation schemes have been developed, varying from domain-specific, such as VERBMOBIL (Alexanderssony et al., 1997), to domain-independent, such as DAMSL (Allen and Core, 1997; Core and Allen, 1997) and DiAML [2] (Bunt et al., 2010, 2012).

Automatically detecting DA labels is an essential step towards describing the discourse structure of conversation (Jurafsky, Shriberg, and Biasca, 1997). DAs are very useful annotations to a large variety of spoken language understanding tasks, such as utterance clustering

1. https://bitbucket.org/guokan_shang/da-classification
2. accepted to be included in the ISO 24617-2 standard. https://www.iso.org/standard/76443.html

Change	Speaker	Utterance	DA
-	B	Of course I use,	sd
True	A	<laughter>.	x
True	B	credit cards.	+
False	B	I have a couple of credit cards	sd
True	A	Yeah.	b
True	B	and, uh, use them.	+
True	A	Uh-huh,	b
False	A	do you use them a lot?	qy
True	B	Oh, we try not to.	ng

Table 4.1 – Fragment from SwDA conversation sw3332. Statement-non-opinion (sd), Non-verbal (x), Interruption (+), Acknowledge/Backchannel (b), Yes-No-Question (qy), Negative non-no answers (ng).

(Shang et al., 2019), real-time information retrieval (Meladianos et al., 2017), conversational agents (Higashinaka et al., 2014; Ahmadvand, Choi, and Agichtein, 2019), and summarization (Shang et al., 2018).

It is difficult to predict the DA of a single utterance without having access to the other utterances in the context. For instance, for an utterance such as "`Yeah`", it is hard to tell whether the associated DA should be 'Agreement', 'Yes answer' or 'Backchannel'. Plus, different labels have different transition probabilities to other labels. E.g., an initial greeting DA is very likely to be followed by another greeting DA. Likewise, a question DA is more likely to be followed by an answer DA. To summarize, it is necessary for a DA classification model to capture dependencies both at the utterance level and at the label level. Recent works (Li and Wu, 2016; Liu et al., 2017; Tran, Zukerman, and Haffari, 2017; Chen et al., 2018; Kumar et al., 2018; Li et al., 2019b; Raheja and Tetreault, 2019) treat DA classification as a sequence labeling problem. The BiLSTM-CRF model (Huang, Xu, and Yu, 2015; Lample et al., 2016), originally introduced for the tasks of POS tagging, chunking and named entity recognition, is the most widely used architecture. In it, a bidirectional recurrent neural network with LSTM cells is first applied to capture the dependencies among consecutive utterances, and then, a Conditional Random Field (CRF) layer is used to capture the dependencies among consecutive DA labels.

CRF is a discriminative probabilistic graphical framework (Koller and Friedman, 2009; Sutton, McCallum, et al., 2012) used to label sequences (Lafferty, McCallum, and Pereira, 2001). It models the conditional probability of a target label sequence given an input sequence. General CRF can essentially model any kind of graphical structure to capture arbitrary dependencies among output variables. For NLP se-

quence labeling tasks, linear chain CRF is the most common variant. The labels are arranged in a linear chain, i.e., only neighboring labels are dependent (first-order Markov assumption). The BiLSTM-CRF architecture employs a linear chain CRF. Hence, for brevity, in the rest of this chapter, the term CRF is short for linear chain CRF.

Recently, neural versions of the CRF have been developed mainly for NLP sequence labeling tasks (Collobert et al., 2011; Huang, Xu, and Yu, 2015; Lample et al., 2016). While traditional CRF requires defining a potentially large set of handcrafted feature functions (each weighted with a parameter to be trained), neural CRF has only two parameterized feature functions (emission and transition) that are trained with the other parameters of the network in an end-to-end fashion.

4.2 MOTIVATION

Most sequence labeling tasks in NLP, such as POS tagging, chunking, and named entity recognition, involve only two sequences: input and target. In DA classification however, we have access to an additional input sequence, that of speaker-identifiers. This extra input could, in principle, greatly improve DA prediction. Indeed, research on turn management (Sacks, Schegloff, and Jefferson, 1974) has shown that dialogue participants do not start or stop speaking arbitrarily, but follow an underlying turn-taking system to occupy or release the speaker role (Petukhova and Bunt, 2009). For instance, the last two utterances in Table 4.1 illustrate a non-arbitrary change of speakers, following a turn-allocation action (here, a question). In this conversational situation, speaker B has to take the turn, to respond to speaker A. To sum up, the sequences of DAs and speakers are tightly interconnected.

However, state-of-the-art DA classification models ignore the sequence of speaker-identifiers (Chen et al., 2018; Kumar et al., 2018; Li et al., 2019b; Raheja and Tetreault, 2019). This is a clear limitation. To address this limitation, we propose in this work a simple modification of the CRF layer where the label transition matrix is conditioned on speaker-change. We evaluate our modified CRF layer within the BiLSTM-CRF architecture, and find that on the SwDA corpus, it improves performance compared to the original CRF. Furthermore, visualizations demonstrate that sophisticated transition patterns between DA label pairs, conditioned on speaker-change, can be learned in an end-to-end way.

4.3 RELATED WORK

In this section, we first introduce the two major DA classification approaches, and then focus on previous work involving the use of BiLSTM-CRF and speaker information.

Multi-class classification. In this first approach, consecutive DA labels are considered to be independent. The DA label of each utterance is predicted in isolation by a classifier such as, e.g., naive Bayes (Grau et al., 2004), Maxent (Ang, Liu, and Shriberg, 2005; Venkataraman et al., 2005), or SVM (Liu, 2006). Since the first application of neural networks to DA classification by Ries (1999), deep learning has shown promising results even with some simple architectures (Khanpour, Guntakandla, and Nielsen, 2016; Shen and Lee, 2016). More recent work developed more advanced models, and started taking into account the dependencies among consecutive utterances (Kalchbrenner and Blunsom, 2013; Lee and Dernoncourt, 2016; Ortega and Vu, 2017; Bothe et al., 2018). For example, in Bothe et al. (2018), the representations of the current utterance and the three preceding utterances are fed into a RNN, and the last annotation is used to predict the DA label of the current utterance.

Sequence labeling. In the second approach, the DA labels for all the utterances in the conversation are classified together. Traditional work uses statistical approaches such as Hidden Markov Models (HMM) (Stolcke et al., 2000; Surendran and Levow, 2006; Tavafi et al., 2013) and CRFs (Lendvai and Geertzen, 2007; Zimmermann, 2009; Kim, Cavedon, and Baldwin, 2010) with handcrafted features. In HMM based approaches, the DA labels are hidden states and utterances are observations emanating from these states. The hidden states are evolving according to a discourse grammar, which essentially is an n-gram language model trained on DA label sequences. Following advances in deep learning, neural sequence labeling architectures (Huang, Xu, and Yu, 2015; Reimers and Gurevych, 2017; Yang, Liang, and Zhang, 2018; Cui and Zhang, 2019; Chapuis et al., 2020; Colombo et al., 2020) have set new state-of-the-art performance. Two major architectures have been tested: BiLSTM-Softmax (Li and Wu, 2016; Liu et al., 2017; Tran, Zukerman, and Haffari, 2017) and BiLSTM-CRF. This study focuses on the BiLSTM-CRF architecture.

BiLSTM-CRF. Kumar et al. (2018) were the first to introduce the BiLSTM-CRF architecture for DA classification. Their model is hierarchical and consists of two levels, where at level 1, the text of each utterance is separately encoded by a shared bidirectional LSTM (BiLSTM) with last-pooling, resulting in a sequence of vectors. At level 2, that sequence is passed through another BiLSTM topped by a CRF layer. At test time, the optimal output label sequence is retrieved from the trained model by Viterbi algorithm (Viterbi, 1967). Chen et al. (2018) and Raheja and Tetreault (2019) improved on the previous model by adding different attention mechanisms. Li et al. (2019b) discovered that performing topic classification as an auxiliary task, can assist in predicting DA labels. The topic of each utterance is automatically determined using Latent Dirichlet Allocation (Blei, Ng, and Jordan, 2003). Their model con-

sists of two BiLSTM-CRF architectures for predicting simultaneously the target DA label sequence and the target topic label sequence. This model represents the state-of-the-art in DA classification.

Speaker information. There are only a few previous works that consider the sequence of speaker-identifiers for DA classification. In Bothe et al. (2018), the utterance representation is the concatenation of the one-hot encoded speaker-identifier, e.g., A as [1, 0] and B as [0, 1], with the output of the RNN-based character-level utterance encoder. By contrast, Li and Wu (2016) and Liu et al. (2017) choose to concatenate the speaker-change vector with the representation obtained via their CNN-based and RNN-based word-level utterance encoders. Speaker-change is binary as shown in Table 4.1, obtained by checking if the current utterance is from the same or different speaker as the previous one. Venkataraman et al. (2005) also include speaker-change as one of the handcrafted features for their Maxent classifier.

Apart from the naive concatenation approaches described above, Kalchbrenner and Blunsom (2013) proposed to let the recurrent and output weights of the RNN cell be conditioned on speaker-identifier, i.e., a speaker-aware RNN cell. Stolcke et al. (2000) proposed to train different discourse grammars for different speakers, to guide DA label transitions in HMM.

4.4 MODEL AND OUR CONTRIBUTION

We first describe the general BiLSTM-CRF model for DA classification, shown in Figure 4.1. Then, in the second subsection, we present our modification of the CRF layer that takes speaker-change into account.

4.4.1 *BiLSTM-CRF model*

Notation. Let us denote by $\{(\mathbf{x}^t, y^t)\}_{t=1}^T$ a conversation of length T. $X = \{\mathbf{x}^t\}_{t=1}^T$ is the sequence of utterances, where each utterance $\mathbf{x}^t = \{x_n^t\}_{n=1}^N$ is itself a sequence of words of length N. $Y = \{y^t\}_{t=1}^T$ denotes the target sequence, where $y^t \in \mathcal{Y}$ is the set of all possible DA labels of size $|\mathcal{Y}| = K$. We use y^t to denote the label and its integer index interchangeably.

Utterance encoder. Each utterance is separately encoded by a shared forward RNN with LSTM cells. Only the last annotation $\mathbf{u}_N^t$ is retained (last pooling). We are left with a sequence of utterance embeddings $\{\mathbf{u}^t\}_{t=1}^T$.

BiLSTM layer. The sequence of utterance embeddings $\{\mathbf{u}^t\}_{t=1}^T$ is then passed on to a bidirectional LSTM, returning the sequence of conversation-level utterance representations $\{\mathbf{v}^t\}_{t=1}^T$.

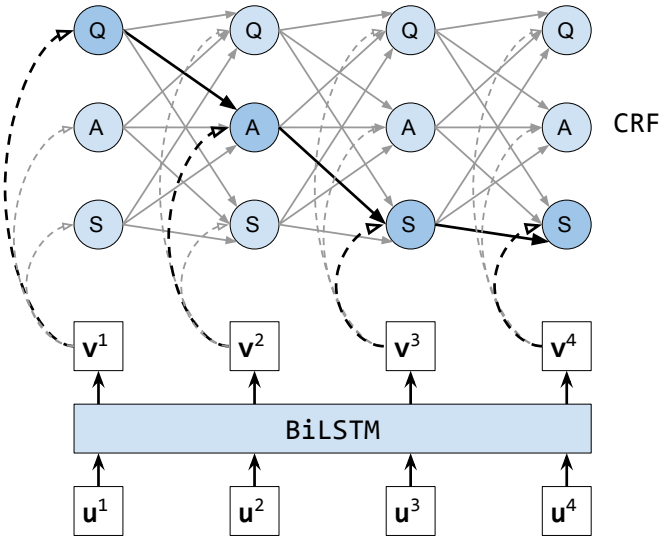

Figure 4.1 – BiLSTM-CRF, for an example with $\{\mathbf{u}^t\}_{t=1}^{T=4}$ (utterance embeddings) as input and Q, A, S, S (DA labels) as target. Three possible labels $\{Q, A, S\}$ stand for Question, Answer, and Statement, respectively.

CRF layer. $\mathbf{v}^t$ can already be used to predict locally the label at each time step in isolation, through a dense layer with softmax activation, which results in the BiLSTM-Softmax architecture. However this might lead to a non-optimal global solution, if we consider the output DA label sequence as a whole.

On the other hand, CRF models the conditional probability $P(Y|X)$ of an entire target sequence Y given an entire input sequence X. Thus, it guarantees an optimal global solution, under the first order Markov assumption. More precisely:

$$P(Y|X) = \frac{\exp(\psi(X, Y))}{\sum_{\tilde{Y}} \exp(\psi(X, \tilde{Y}))} \tag{4.1}$$

where $\psi(X, Y)$ is a feature function that assigns a *path score* to the label sequence Y, giving X. Then, a softmax function is used to yield the conditional probability, where $\tilde{Y}$ denotes one of all possible label sequences (paths).

$\psi(X, Y)$ is defined as the sum of *emission scores* (or state scores) and *transition scores* over all time steps (Morris and Fosler-Lussier, 2006; Chen and Moschitti, 2019):

$$\psi(X, Y) = \sum_{t=1}^{T} h(y^t, X) + \sum_{t=1}^{T-1} g(y^t, y^{t+1}) \tag{4.2}$$

Emission (state) scores are assigned to the dashed top-down edges (nodes) in Figure 4.1, computed as follows:

$$h(y^t, X) = (\mathbf{W}\mathbf{v}^t + \mathbf{b})[y^t] \tag{4.3}$$

where the conversation-level utterance representation $\mathbf{v}^t$ is converted into a vector of size K and $[y^t]$ denotes the element at index y^t. Higher values of $h(y^t, X)$ indicate that the model is more confident in predicting the output label y^t at time step t.
Transition scores are assigned to the solid left-to-right edges in Figure 4.1, computed as follows:

$$g(y^t, y^{t+1}) = \mathbf{G}[y^t, y^{t+1}] \tag{4.4}$$

where $\mathbf{G}$ is the label transition matrix of size $K \times K$. E.g, the element $\mathbf{G}[y^t, y^{t+1}]$ is the transition score from label y^t to label y^{t+1}. Note that the transition matrix is shared across all time steps.

The CRF layer is parameterized by $\mathbf{W}$, $\mathbf{b}$, and $\mathbf{G}$. To learn these parameters and those of the previous layers, maximum likelihood estimation is used. For a training set of M conversations, the loss can be written as:

$$\mathcal{L} = \sum_{m=1}^{M} -\log P(Y^m|X^m) \tag{4.5}$$

At test time, the optimal output label sequence, i.e., $Y^* = \text{argmax}_{\tilde{Y}} P(\tilde{Y}|X)$ for unseen X, is obtained with the Viterbi decoding algorithm (Viterbi, 1967). Due to the Markov property of the linear chain CRF, the computations of Viterbi algorithm and the normalization term in Equation 4.1 can be broken down into a series of sub-problems over time in a recursive manner, which are solved via dynamic programming (Bellman, 1966), with polynomial complexity $O(TK^2)$.

4.4.2 *Our contribution*

Given the sequence of speaker-identifiers $S = \{s^t\}_{t=1}^T$, we can instantly derive the sequence of speaker-changes $Z = \{z^{t,t+1}\}_{t=1}^{T-1}$ by comparing neighbors. E.g., $z^{2,3} = 0$ means the speaker does not change from time $t = 2$ to $t = 3$.

We extend the original CRF so that it considers as additional input, the sequence Z. That is, CRF now models $P(Y|X, Z)$ instead of just $P(Y|X)$. In other words, the prediction of the DA label sequence is now conditioned both on the utterance sequence and the speaker-change sequence. Specifically, transition scores in our modified CRF layer are computed as follows:

$$\begin{aligned} g(y^t, y^{t+1}, z^{t,t+1}) =& (1 - z^{t,t+1}) * \mathbf{G}_0[y^t, y^{t+1}] + \\ & z^{t,t+1} * \mathbf{G}_1[y^t, y^{t+1}] \end{aligned} \tag{4.6}$$

where $\mathbf{G}_0$ and $\mathbf{G}_1$ are label transition matrices of size $K \times K$, corresponding respectively to the "speaker unchanged" and "speaker changed" cases.

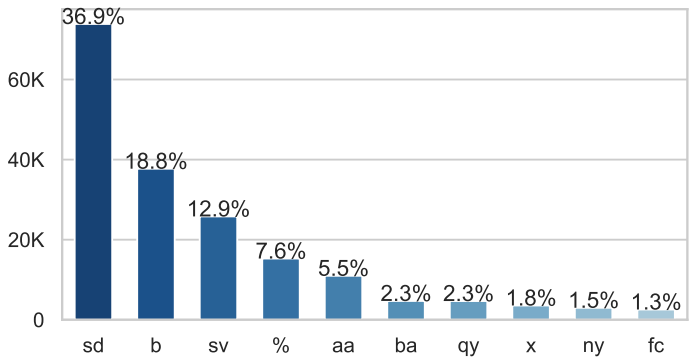

Figure 4.2 – Counts and frequencies of the 10 most represented DA labels in the SwDA dataset.

4.5 EXPERIMENTAL SETUP

Dataset. We experiment on the widely-used SwDA [3] (Switchboard Dialogue Act) dataset (Jurafsky, Shriberg, and Biasca, 1997; Stolcke et al., 2000). This corpus contains telephone conversations recorded between two randomly selected speakers talking about one of various general topics (air pollution, music, football, etc.). In this dataset, utterances are annotated with 42 mutually exclusive DA labels, based on the SWBD-DAMSL annotation scheme (Jurafsky, Shriberg, and Biasca, 1997). Inter-annotator agreement is 84%. The frequency of the 10 most represented DA labels are illustrated in Figure 4.2. We can see that labels are highly imbalanced and follow a long-tailed distribution. Detailed statistics for all 42 labels are provided in Table 4.2.

We adopt the same training, validation and testing partition as previous work (Lee and Dernoncourt, 2016) [4], consisting of 1003, 112, and 19 conversations, respectively.

A note about the '+' tag. The '+' tag, as shown in Table 4.1, accounts for 8.1% of the total annotations, but is not part of the default label set. That tag is used to mark the remaining parts of an utterance that has been interrupted by the other speaker. While most of the previous works did not predict, or even mention this tag, some efforts considered it as a 43^{rd} DA label and predicted it (Lee and Dernoncourt, 2016; Raheja and Tetreault, 2019).

In this work, we followed the approach of (Webb, Hepple, and Wilks, 2005; Milajevs and Purver, 2014; Kim et al., 2017), and used the '+' tag to reconnect, bottom-up, all the parts of an interrupted utterance together. E.g., in Table 4.1, the parts "`Of course I use,`" and "`credit cards.`", uttered by speaker B, are reconnected into "`Of course I use, credit cards.`", which becomes the new first utterance. It is followed

3. https://github.com/cgpotts/swda
4. https://github.com/Franck-Dernoncourt/naacl2016

Dialogue Act (label)	count	frequency	Dialogue Act (label)	count	frequency
Statement-non-opinion (sd)	73873	36.85%	Collaborative Completion (^2)	709	0.35%
Acknowledge/Backchannel (b)	37727	18.82%	Repeat-phrase (b^m)	677	0.34%
Statement-opinion (sv)	25810	12.88%	Open-Question (qo)	647	0.32%
Abandoned/Uninterpretable (%)	15294	7.63%	Rhetorical-Questions (qh)	566	0.28%
Agree/Accept (aa)	10987	5.48%	Hold before answer/agreement (^h)	546	0.27%
Appreciation (ba)	4702	2.35%	Reject (ar)	341	0.17%
Yes-No-Question (qy)	4679	2.33%	Negative non-no answers (ng)	296	0.15%
Non-verbal (x)	3565	1.78%	Signal-non-understanding (br)	295	0.15%
Yes answers (ny)	2995	1.49%	Other answers (no)	284	0.14%
Conventional-closing (fc)	2562	1.28%	Conventional-opening (fp)	225	0.11%
Wh-Question (qw)	1954	0.97%	Or-Clause Question (qrr)	208	0.10%
No answers (nn)	1363	0.68%	Dispreferred answers (arp_nd)	207	0.10%
Response Acknowledgement (bk)	1299	0.65%	3rd-party-talk (t3)	115	0.06%
Hedge (h)	1204	0.60%	Offers, Options, Commits (oo_co_cc)	109	0.05%
Declarative Yes-No-Question (qy^d)	1203	0.60%	Maybe/Accept-part (aap_am)	105	0.05%
Backchannel in question form (bh)	1036	0.52%	Self-talk (t1)	103	0.05%
Quotation (^q)	948	0.47%	Downplayer (bd)	101	0.05%
Summarize/reformulate (bf)	928	0.46%	Tag-Question (^g)	92	0.05%
Other (ot)	876	0.44%	Declarative Wh-Question (qw^d)	80	0.04%
Affirmative non-yes answers (na)	841	0.42%	Apology (fa)	78	0.04%
Action-directive (ad)	740	0.37%	Thanking (ft)	74	0.04%

Table 4.2 – Counts and frequencies of the 42 DA labels in the SwDA dataset. There are 200444 utterances in total.

by "<laughter>", uttered by speaker A. We opted for this approach as predicting the DA of a broken utterance sometimes does not make sense. For instance, in this situation with three utterances: (1) "A: so, (Wh-Question)", (2) "B: <throat_clearing> (Non-verbal)", and (3) "A: what's your name? (+)", it is very difficult to correctly predict that utterance 1 is a question. And predicting anything other than a question-related tag for utterance 3 does not really make sense. Reconstructing 1 and 3 into a single utterance "A: so, what's your name? (Wh-Question)" solves both issues.

Implementation and training details. Disfluency markers (Meteer et al., 1995) were filtered out and all characters converted to lowercase. We used some optimal hyperparameters provided by Kumar et al. (2018). E.g., 0.2 dropout was applied to the utterance embeddings and conversation-level utterance representations, and all LSTM layers had 300 hidden units. The embedding layer was initialized using 300-dimensional word vectors pre-trained with the gensim (Řehůřek and Sojka, 2010b) implementation of word2vec (Mikolov et al., 2013a) on the utterances of the training set, and was frozen during training. Vocabulary size was around 21K, and out-of-vocabulary words were mapped to a special token [UNK], randomly initialized.

Models were trained with the Adam optimizer (Kingma and Ba, 2015). Early stopping was used on the validation set with a patience of 5 epochs and a maximum number of epochs of 100. The best epoch was selected as the one associated with the highest validation accuracy. Usually, the best epoch was within the first 10. We set our batch-size to be 1, i.e, one conversation for one training step. Batch sizes of 1, 2, 4, 8, and 16 were also tried, without observing significant differences.

4.6 QUANTITATIVE RESULTS

Performance comparison. Table 4.5 reports the results in terms of classification accuracy, averaged over 10 runs to account for the randomness of SGD. Model a) uses our modified CRF layer. Model b) has exactly the same architecture as a), but uses a vanilla CRF layer.

Results show that, in terms of overall accuracy on the test set of 42 DA labels, our model a) outperforms the base model b) by 1%. Moreover, the small standard deviations highlight the consistency of this improvement over the 10 runs. Note that this performance gain is solely caused by our modified CRF layer capturing speaker-change, and is greater than the gains of 0.26% (Liu et al., 2017) and 0.09% (Bothe et al., 2018) reported by previous attempts at leveraging speaker information.

To interpret the results in more detail, we show in Figure 4.3 the confusion matrices of our model and the base model, for the 10 most frequent DA labels, representing close to 91% of all annotations. The rows

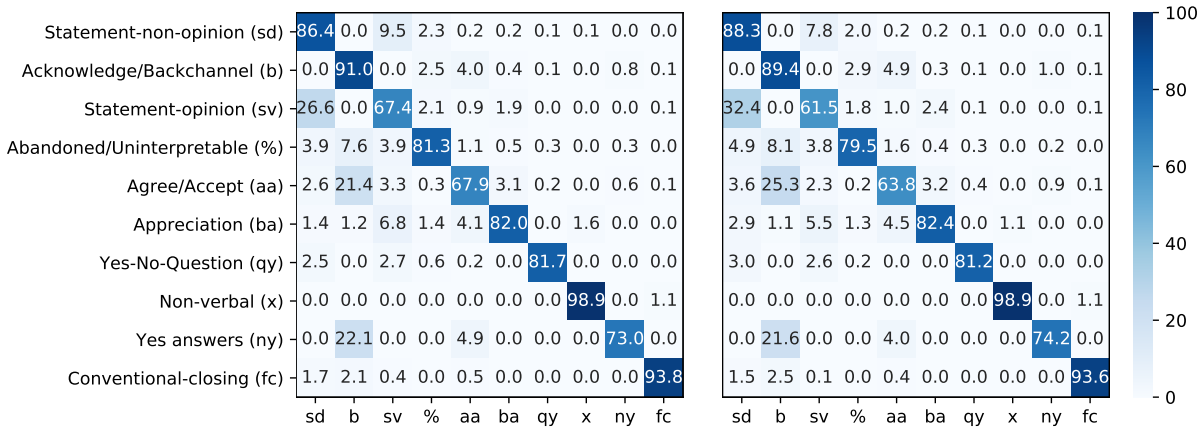

Figure 4.3 – Normalized confusion matrices, averaged over 10 runs, for the 10 most frequent DA labels (90.9% of all annotations). Left: our model, right: base model. Rows (columns) correspond to true (predicted) classes.

correspond to true classes, and the columns to predicted classes. [5] By looking at the diagonals, we can see that our model (on the left) better predicts 6 labels out of 10 with absolute accuracy gains of up to 5.9% (for statement-opinion, `sv`) [6] and is on par with the baseline model for one label (non-verbal, `x`), at 98.9%. By looking at off-diagonal values, miss rates are decreased up to 5.8% (for `sv` misclassified as `sd`) by our model. Also, our model provides a large boost for the (Acknowledge/Backchannel, Agree/Accept), or (`b`, `aa`) pair. It increases the respective accuracies by 1.6% (89.4%→91.0%) and 4.1% (63.8%→67.9%). The respective miss rates are decreased by 0.9% (4.9%→4.0%) and 3.9% (25.3%→21.4%), respectively for `b` misclassified as `aa` and `aa` misclassified as `b`. This is to be noted, as these two labels are among the most frequently confused pairs (Chen et al., 2018; Kumar et al., 2018).

Although our model achieves significant gains on a majority of the most frequent labels, it decreases performance for the most frequent label, `sd`, which accounts for 36.9% of all labels, as shown by Figure 4.2. This explains why, in terms of overall accuracy, our improvements are modest. In addition, the performance drop regarding `sd` can be interpreted as a consequence of the trade-off between `sd` and `sv`, since the distinction between them was very hard to make even by annotators (Jurafsky, Shriberg, and Biasca, 1997). This can be demonstrated in terms of precision, recall, and F1 score, as shown in Table 4.3. We can observe that, as opposed to the base model, our model has lower `sd` and higher `sv` recall values. A similar observation can be made for

5. Note that our confusion matrices were row-wise normalized by class size. So we use the terms accuracy (per class) to denote diagonal values (equivalent to recall or hit rate), and miss rate for off-diagonal values.

6. The margins are even larger (up to 20%) on some less frequent labels, as shown by the results in Section 4.7.

		P	R	F1
Our	sd	80.49	86.36	83.32
	sv	71.54	67.41	69.42
Vanilla	sd	77.83	88.32	82.74
	sv	73.24	61.48	66.84

Table 4.3 – Precison, Recall, and F1 score (%) of our model vs. base model on the `sd` and `sv` labels.

	Ours	Vanilla	Diff.
10 best DAs	37.08	31.70	+ 5.38
10 worst DAs	59.67	64.54	- 4.87

Table 4.4 – Average accuracy (%) of our model vs. base model on the 10 DAs best and worst predicted by our model (resp. representing 20% and 40% of all annotations).

precision scores. Thus, the prediction between `sd` and `sv` is a trade-off made by models. It is also interesting to note that our model is superior for both labels in terms of F1 score.

We can observe in Table 4.4 that our model brings improvement where it is most necessary, i.e., for the most difficult and rare DAs (20%). Full details are provided in Section 4.7, along with the corresponding confusion matrices.

The benefits of considering speaker information vary across DA labels. Our model and the base model performed very closely on 4 labels: Non-verbal (`x`), Conventional-closing (`fc`), Appreciation (`ba`), and Yes-No-Question (`qy`). We found that the utterances of these labels contain clear lexical cues that can be mapped to corresponding DA labels in a *non-ambiguous* way. Some examples include "`<laughter>`" → `x`, "`Bye-bye.`" → `fc`, "`That's great.`" → `ba`, and "`Do you …?`" → `qy`. In other words, predicting well these four DAs does not require having access to speaker information. It can be done solely from the text of the current utterance. Having access to context is not even required. This explain why our speaker-aware CRF is not helpful here.

This interpretation is supported by the fact that, as explained in Section 4.7, our model is most useful for the DAs that require speaker-change awareness.

Ensembling and joint training. Since the model using our CRF and the model using the vanilla CRF appear to have their own strengths and weaknesses, we tried combining them to improve performance. More precisely, we experimented with two approaches. First, an ensembling approach that combines the predictions of the two trained

	Model	BiLSTM input	CRF extra input	Accuracy (% ± SD)
a)	Our CRF	$\mathbf{u}^t$	SC	**78.70** ± .37
a1)		$\mathbf{u}^t$ + SI	SC	78.32 ± .28
a2)		$\mathbf{u}^t$ + SC	SC	78.65 ± .47
b)	Vanilla CRF	$\mathbf{u}^t$	-	77.69 ± .38
b1)		$\mathbf{u}^t$ + SI	-	77.86 ± .61
b2)		$\mathbf{u}^t$ + SC	-	78.33 ± .71
c)	Softmax	$\mathbf{u}^t$	-	77.80 ± .48
c1)		$\mathbf{u}^t$ + SI	-	77.73 ± .44
c2)		$\mathbf{u}^t$ + SC	-	78.33 ± .49
a) + b)	ensembling	$\mathbf{u}^t$	SC	**78.89** ± .20
a) + b)	joint training	$\mathbf{u}^t$	SC	78.27 ± .47

Table 4.5 – Results, averaged over 10 runs. SI: speaker-identifier, SC: speaker-change, $\mathbf{u}^t$: utterance embedding, ±: standard deviation.

models by averaging their emission and transition scores (respectively). Second, a joint training approach that combines the two models into a new one and trains it from scratch. In that second model, our CRF and the vanilla CRF are combined, and transition scores are computed as:

$$g(y^t, y^{t+1}, z^{t,t+1}) = \mathbf{G}_{basis}[y^t, y^{t+1}] + \\ (1 - z^{t,t+1}) * \mathbf{G}_0[y^t, y^{t+1}] + z^{t,t+1} * \mathbf{G}_1[y^t, y^{t+1}] \quad (4.7)$$

where G_{basis} is the transition matrix as in the original CRF, used at each time step, while $\mathbf{G}_0/\mathbf{G}_1$ are applied only when the speaker does not change/changes, as in our modified CRF layer.

Results in Table 4.5 show that the ensemble model reaches new best performance at 78.89, providing close to a 0.2 boost from our model, and a 1.2 boost from the vanilla CRF model. On the other hand, the jointly-trained model does not outperform our model. After inspecting the transition matrices for the two cases ($\mathbf{G}_{basis}$ + $\mathbf{G}_0$) and ($\mathbf{G}_{basis}$ + $\mathbf{G}_1$), we found that the addition of $\mathbf{G}_{basis}$ blurred the label transition patterns.

Ablation studies. Our results showed that considering speaker information improves DA classification. Then, we wanted to confirm whether our way of taking speaker information into account (at the CRF level) was the most effective. To this purpose, we trained two

other base models, both using the vanilla CRF. These two models respectively concatenate the one-hot encoded speaker-identifier vector (noted SI, of size 2) and the binary speaker-change vector (noted SC, of size 1) with the utterance embedding $\mathbf{u}^t$.[7] Results, shown in rows b1 and b2 of Table 4.5, show that while they bring improvement compared to the basic base model (row b), these two approaches are not able to yield as big of a gain as the model using our modified CRF layer, indicating that taking speaker-change into account at the CRF level is superior.

For the sake of completeness, we repeated these experiments with our model. Results, available in rows a1 and a2 of Table 4.5, show that performance was not improved (78.32 and 78.65 vs. 78.70). Thus, it seems that taking speaker information into account twice, both at the BiLSTM level and at the CRF level, is not useful, or at least, not in this way.

Results in Table 4.5 also show that SC is a better feature than SI in general.

BiLSTM-CRF vs. BiLSTM-Softmax. To the best of our knowledge, no previous study has compared BiLSTM-CRF to BiLSTM-Softmax on the DA classification task. Hence, in this work, we decided to compare between these two models. Results reveal that the models using BiLSTM-Softmax (rows c, c1, and c2) are competitive with the ones using BiLSTM-CRF (rows b, b1, and b2). More specifically, BiLSTM-Softmax outperforms BiLSTM-CRF with text features only (rows b vs. c), by a slight 0.11 margin, but it is the opposite for text + SI (b1 vs. c1, 0.13 difference). With text + SC (b2 vs. c2), they achieve similar performance.

These results are not very surprising, since, on other tasks than DA classification, multiple recent works have reported that BiLSTM-CRF does not always outperform BiLSTM-Softmax (Reimers and Gurevych, 2017; Yang, Liang, and Zhang, 2018; Cui and Zhang, 2019). For example, in Yang, Liang, and Zhang (2018), CRF brought improvement for named entity recognition and chunking, but not for POS tagging. One of the reasons might be that the simple Markov label transition model of CRF does not give much information gain over strong neural encoding (Cui and Zhang, 2019). That is, BiLSTM may be expressive enough to implicitly capture the *obvious* dependencies among labels.

In any case, the model equipped with our CRF layer (row a) outperforms all variants of BiLSTM-Softmax and BiLSTM-vanilla_CRF. This suggests that our CRF layer can capture richer and *not obvious* label dependencies given speaker information, which, in the end, makes the use of a CRF layer valuable in assisting DA classification.

7. proposed in Liu et al. (2017) and Bothe et al. (2018), but not in the context of BiLSTM-CRF.

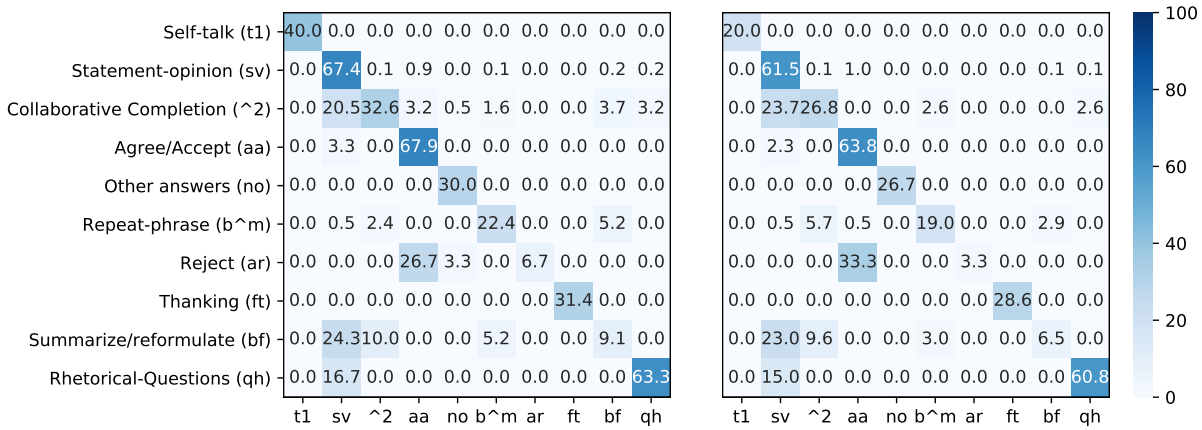

Figure 4.4 – Normalized confusion matrices, averaged over 10 runs, for the 10 DA labels **best** predicted by our model (20.2% of all annotations). Left: our model, right: base model.

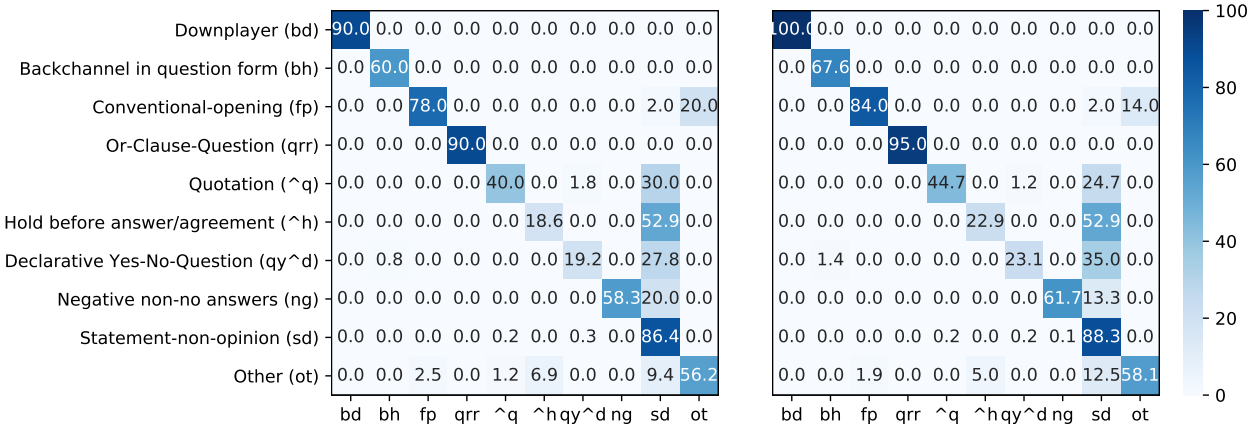

Figure 4.5 – Normalized confusion matrices, averaged over 10 runs, for the 10 DA labels **worst** predicted by our model (39.6% of all annotations). Left: our model, right: base model.

4.7 WORST AND BEST CASE ANALYSIS

In addition to the comparison provided in the previous section for the 10 most frequent DA labels. In this section, we interpret the confusion matrices for the 10 DA labels that our model best predicted (Figure 4.4) and worst predicted (Figure 4.5), in comparison with the base model, always on the right. Inspecting the matrices reveals that our model is most useful for the DAs requiring speaker-change awareness, which confirms the effectiveness of our modification of the CRF layer. It also shows that our model brings improvement where it is most necessary, i.e., for the most difficult and rare DAs.

Relative differences. For the 10 DA labels best predicted by our model, the average performance gain compared to the base model is equal to

5.38 (shown in Table 4.4), whereas the drop in performance for the 10 DAs worst predicted by our model is lower, only equal to **4.87**. Thus, when it improves performance, our model does so with a greater margin than when it decreases performance. This fact is hidden when simply looking at the global accuracy over the 42 DA labels, because the 10 best DAs for our model only correspond to 20.2% of all annotations, whereas the 10 worst account for almost 40% of all annotations.

Absolute differences. It is also interesting to note that the 10 DAs that our model best predicted are all very difficult DAs, for which the performance of the base model is very low in the first place: **31.7**, on average. These DAs are also rare: they only correspond to 20.2% of all annotations. Our model raises the average accuracy on these labels to **37.08**. On the other hand, the 10 DAs that are worst predicted by our model are more frequent DAs (40% of all annotations), for which the performance of the base model is already quite high: **64.54**, on average. And although our model is not as good as the base model on these DAs, it still reaches a decent average performance of **59.67**. Therefore, our model provides a performance boost where it is most necessary (difficult and rare DAs), and wherever it fails, it still provides decent accuracy levels.

10 best DAs for our model. Our model outperforms the base model by a very large margin of 20.0% (20.0%→40.0%) for Self-talk (`t1`, the speaker talks to him/herself). It makes a lot of sense, as the accurate prediction of this DA obviously requires being aware of speaker-change. Similar conclusions can be also drawn for Collaborative Completion (`^2`, one speaker completes the other speaker's utterance), Repeat-phrase (`b^m`, repeating parts of what the previous speaker said), Thanking (`ft`), Summarize/reformulate (`bf`, proposing a summarization or paraphrase of another speaker's talk/point), and Rhetorical-Questions (`qh`, questions asked to make a statement or asked to produce an effect with no answer expected).

10 worst DAs for our model. On the other hand, speaker information does not seem to be crucial to predict the 10 DA labels most often missed by our model. For instance, Conventional-openings (fp) are always found among the first utterances in a conversation, so there is only a small need for speaker-change awareness in that case. E.g., in this situation with three utterances: (1) "`A: Hi, Wanet (fp)`", (2) "`A: How are you? (fp)`", and (3) "`B: I'm doing fine. (fp)`", utterances 2 and 3 are labeled with `fp`, regardless of speaker-change. Likewise, the need for speaker-change awareness seems very little for the Quotation (^q) and Other (ot) DAs. In other words, among the DAs worst predicted by our model are DAs for which speaker information is not necessary to make an accurate prediction. This makes sense, since the

goal of our modified CRF layer is precisely to capture speaker information.

4.8 QUALITATIVE RESULTS

Visualization of transition matrices. We illustrate, in Figure 4.6, the transition matrices $\mathbf{G}_0$ and $\mathbf{G}_1$ of our CRF layer, together with the single matrix $\mathbf{G}$ of the vanilla CRF layer. This visualization is done for 12 labels that are easy to interpret, such as statements, questions, answers, etc. We can observe some interesting patterns, sometimes matching intuition, and sometimes harder to interpret. We report some of the most interesting findings below:

— Overall, $\mathbf{G}_0$ and $\mathbf{G}_1$ are not identical, which means that different transition patterns are associated with the "speaker unchanged" and "speaker changed" cases. The dark diagonal of $\mathbf{G}_0$ shows that when the speaker does not change, the majority of labels tend to carry over to the next utterance. On the opposite, $\mathbf{G}_1$ clearly shows that changing speakers very often induce a change in DA.

— questions starting with words including: 'what', 'how', etc. (`qw` label) tend to transfer to statements (`sd` and `sv`) and to other answers (`no`, e.g., "`I don't know`") if the speaker changed, but to other forms of questions, yes-no questions and questions starting with the word 'or' (`qy` and `qrr`), or to acknowledgements (`bk` and `b`) if the speaker did not change. This probably corresponds to instances when the same speaker clarifies, elaborates on, or answers, an original question.

— `sv` label (statement with opinion) tends to transition to Agree/Accept `aa` and Reject `ar` if the speaker changed, while no such clear pattern can be observed for the `sd` label (statement without opinion).

— `qy` label (yes/no questions) tend to transfer to answer labels `ny` (yes), `nn` (no), `no` (other) if the speaker changed, and to another type of question (e.g., or-clause) if the speaker did not change. Again, the latter surely corresponds to the case where a given speaker elaborates on his or her original question.

— answer labels (`ny`, `nn`, `no`) tend to be followed by Response Acknowledgement `bk` and Acknowledge/Backchannel `b` if the speaker changed, but by themselves or statements (`sd` and `sv`) if the speaker did not change.

As far as the transition matrix $\mathbf{G}$ of the vanilla CRF layer (right of Figure 4.6), we can observe that it tries to capture, at the same time, the transition patterns of both the "speaker changed" and "speaker unchanged" cases. For example, `sv` equally tends to transfer to `sv`, `aa`

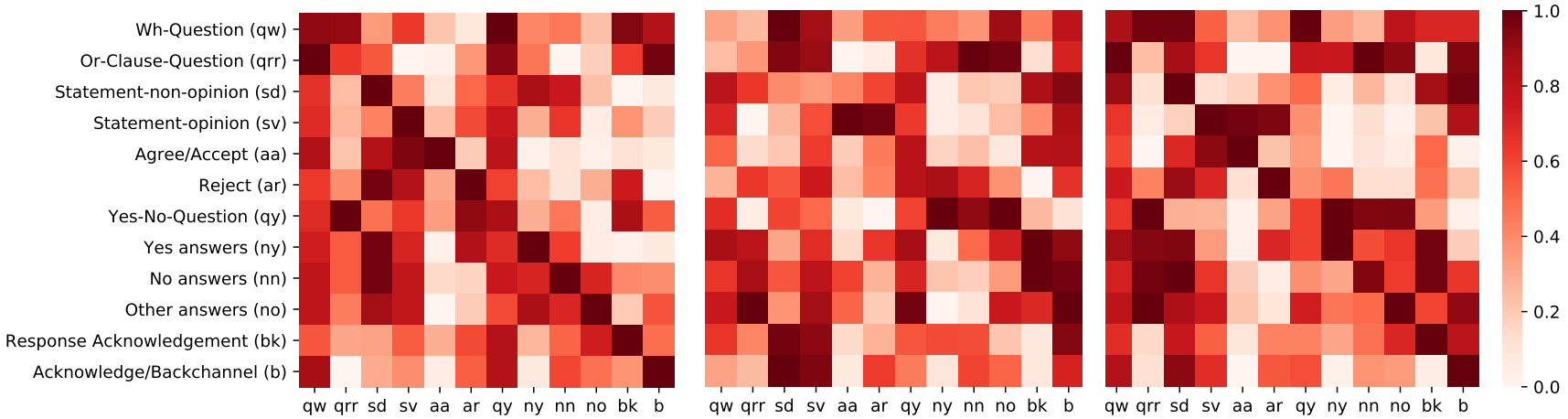

Figure 4.6 – Normalized transition matrices (averaged over 10 runs). Left and center: $\mathbf{G}_0$ (speaker unchanged) and $\mathbf{G}_1$ (speaker changed) of our CRF layer. Right: $\mathbf{G}$ of vanilla CRF layer. The darker, the greater the score.

and ar in **G**, while the transitions towards sv/aa,ar are only probable if the speaker stays the same/changes, as clearly illustrated by $\mathbf{G_0}/\mathbf{G_1}$. Obviously, using two matrices as in our approach gives much more expressiveness to the model in capturing DA label transition patterns. To summarize, visualizations show that the transition matrices $\mathbf{G}_0$ and $\mathbf{G}_1$ in our modified CRF layer are able to encode speaker-change-aware, sophisticated DA transition patterns.

4.9 DISCUSSION AND CONCLUSION

Note that, for our utterance encoder, we also experimented with a bidirectional LSTM (also with last pooling), as in Kumar et al. (2018), and with a bidirectional LSTM with self-attention mechanism (Yang et al., 2016a). However, since they were not giving better results, we opted for the simplest option. One possible explanation for the self-attention mechanism not being helpful could be the very short size of the utterances in the SwDA dataset (68.7% of utterances are shorter than 10 tokens). On such short sequences, a RNN with a 300-dimensional hidden layer is very likely able to keep the full sequence into memory. As far as why a forward RNN suffices, it should be noted that with last pooling, the last time step corresponds to the first annotation of the backward RNN. This is not adding much information to the last annotation of the forward RNN, which represents the entire sequence.

Our goal was not to exceed the state-of-the-art accuracy reported in Li et al. (2019b), Raheja and Tetreault (2019) and Colombo et al. (2020), this is why we used simple models in all of our experiments. However, our improved CRF layer can be directly plugged into more advanced architectures, such as Att-BiLSTM-CRF (Luo et al., 2018) or Transformer-CRF (Chen, Zhuo, and Wang, 2019; Winata et al., 2019; Yan et al., 2019; Zhang and Wang, 2019), and should in principle be able to boost performance regardless of the model used.

In this work, we focused on demonstrating that taking speaker information into consideration was beneficial to the task of DA classification, with the BiLSTM-CRF architecture. We proposed a modified CRF layer that takes as extra input the sequence of speaker-changes. Experiments conducted on the SwDA dataset showed that our CRF layer outperforms vanilla CRF, and brings greater gains than previous attempts at taking speaker information into account. Moreover, visualizations confirmed that our improved CRF was able to learn complex speaker-change aware DA transition patterns in an end-to-end way.

Future research should be devoted to address the limitation of the Markov property of CRF layer, by developing a model that is capable of capturing longer-range dependencies within and among the three sequences: that of speakers, utterances, and DA labels.

5

ABSTRACTIVE COMMUNITY DETECTION

In this chapter, we focus on the task of Abstractive Community Detection, in which utterances in a conversation are grouped according to whether they can be jointly summarized by a common abstractive sentence. Note that in Chapter 3, we approached this task by applying the *k*-means clustering algorithm on TF-IDF vectors of utterances. While this *unsupervised* method was somewhat successful in our previously proposed framework, we felt it could be greatly improved. First, for reasons that become clear if we look at the human-annotated abstractive community examples in Section 5.1, we think that communities should capture more complex relationships than simple lexical/semantic similarity, but this is far beyond the capabilities of TF-IDF and *k*-means. Second, as mentioned earlier in Subsection 1.4, we believe that abstractive community detection plays a crucial role in bridging the gap between extractive and abstractive meeting summarization, but this is little explored in the literature. All these reasons ultimately motivated and drove us to deeply rethink and investigate this task.

This chapter provides a novel *supervised* approach to this task that makes use of human abstractive-extractive linking annotations (see Subsection 2.3 for more details). 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. Moreover, we propose a general sampling scheme that enables the triplet architecture to capture subtle clustering patterns, such as overlapping and nested communities. Experiments on the AMI corpus show that our system outperforms multiple energy-based and non-energy based baselines from the state-of-the-art, and visualization illustrates that our triplet sampling scheme is effective. Code and data are publicly available [1].

5.1 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*. Unlike sentences in traditional documents, utterances are frequently associated with meta-information in the form of discourse features such as speaker ID/role,

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

dialogue act, etc. Utterances are also often ill-formed, incomplete, and ungrammatical, due to the nature of spontaneous communication.

Abstractive summarization of conversations is an open problem in NLP. It requires the machine to gain a high-level understanding of the dialogue, in order to extract useful information and turn it into meaningful abstractive sentences. Previous work (Mehdad et al., 2013; Oya et al., 2014; Banerjee, Mitra, and Sugiyama, 2015; Shang et al., 2018) decomposes this task into two subtasks `a` and `b` as shown in Figure 5.1.

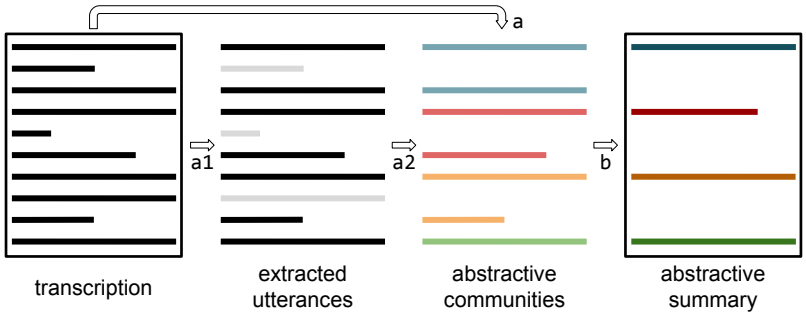

Figure 5.1 – Abstractive community detection: the first step towards summarizing a conversation.

Subtask `a`, or *Abstractive Community Detection* (ACD), is the focus of this work. It consists in grouping utterances according to whether they can be jointly summarized by a common abstractive sentence (Murray, Carenini, and Ng, 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 general problem than, topic clustering. Indeed, as shown in Figure 5.2, 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 nested and overlapping groupings (e.g., D/C and A/D in Figure 5.2, resp.), or be a community of its own, i.e., a singleton community (e.g., B in Figure 5.2).

In this work, we depart from previous work and argue that the ACD subtask should be broken down into two steps, `a1` and `a2` in Figure 5.1. That is, summary-worthy utterances should first be extracted from the transcription (`a1`), and then, grouped into abstractive communities (`a2`). This $a1 \rightarrow a2 \rightarrow b$ 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 Figure 5.2), i.e., create communities. Abstractive summaries comprise four sections: `ABSTRACT`, `ACTIONS`, `PROBLEMS`, and `DECISIONS`.

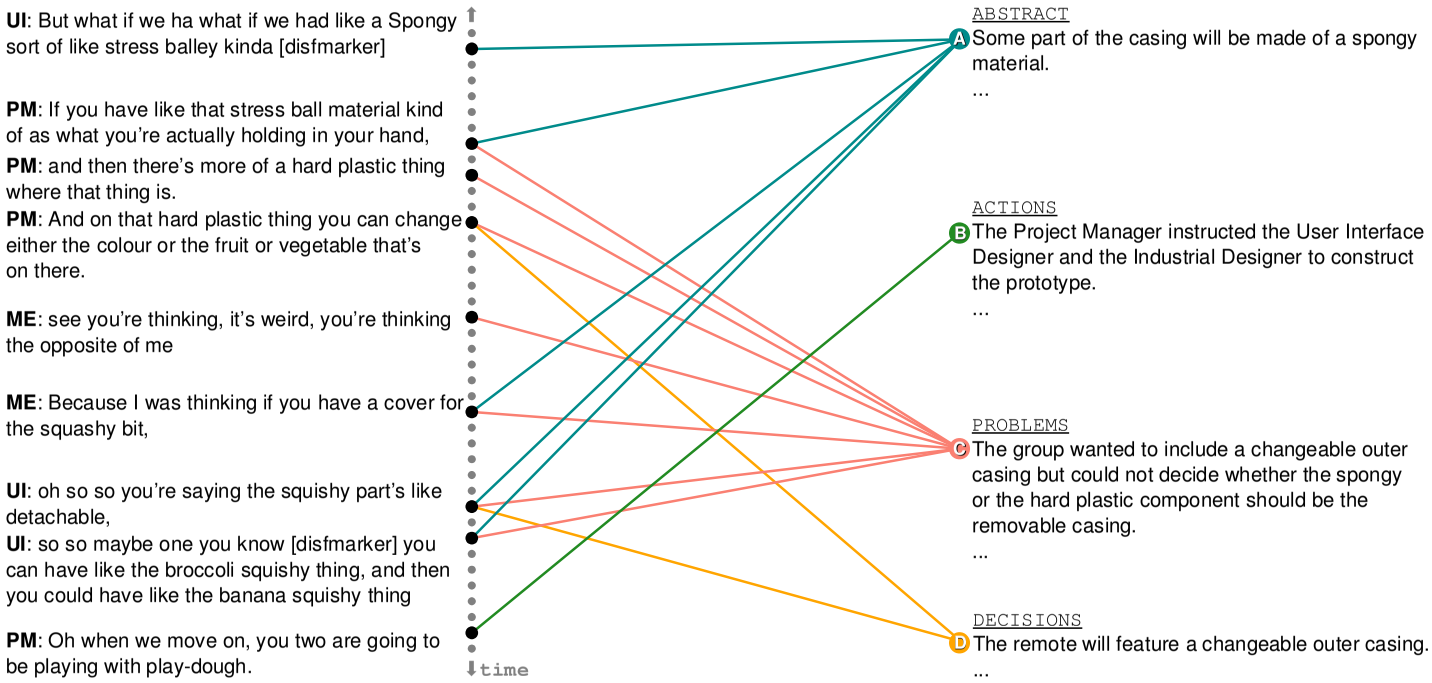

Figure 5.2 – Example of ground truth human annotations from the ES2011c AMI meeting. Successive grey nodes on the left denote utterances in the transcription, where black nodes correspond to the utterances judged important (summary-worthy). Sentences from the abstractive summary are shown on the right. All utterances linked to the same abstractive sentence form one community. Speaker roles are PM: project manager, ME: marketing expert, UI: user interface designer.

Step `a1` plays an important filtering role, since in practice, only a small part of the original utterances are summary-worthy and 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, Renals, and Carletta, 2005; Garg et al., 2009; Tixier, Meladianos, and Vazirgiannis, 2017).

Rather, we focus in this work 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.

5.2 RELATED WORK

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

In the supervised case, Murray, Carenini, and Ng (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, Mitra, and Sugiyama (2015), and Singla et al. (2017) assume that disjoint topic segments (Galley et al., 2003; Eisenstein and Barzilay, 2008) 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, Kalenichenko, and Philbin, 2015) and sentence matching (Mueller and Thyagarajan, 2016), we introduce in this work a novel utterance encoder, and train it within the siamese (Chopra, Hadsell, and LeCun, 2005) and triplet (Hoffer and Ailon, 2015) energy-based meta-architectures. Our final network is able to accurately capture the complexity of abstractive com-

munity 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 multifold:

— 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,

— we propose one of the first applications of energy-based learning to spoken language understanding,

— we introduce a novel utterance encoder featuring three types of self-attention mechanisms and taking contextual and temporal information into account,

— we propose a sampling scheme that enables the triplet architecture to capture subtle levels of similarity such as overlapping and nested clusters. This is a major improvement over prior work, in which only the usual similar/dissimilar case is tackled,

— through extensive experiments, we study the impact of the major components on performance.

5.3 ENERGY-BASED LEARNING

Energy-Based Modeling (EBM) (LeCun and Huang, 2005; Lecun et al., 2006) is a unified framework that can be applied to many machine learning 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] $\mathcal{L}$:

$$W^* = \arg\min_{W \in \mathcal{W}} \mathcal{L}(E_W(X, Y), \mathcal{S}) \tag{5.1}$$

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

5.3.1 *Single architecture*

In the EBM framework, a regression problem can be formulated as shown in Figure 5.3a, 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.

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

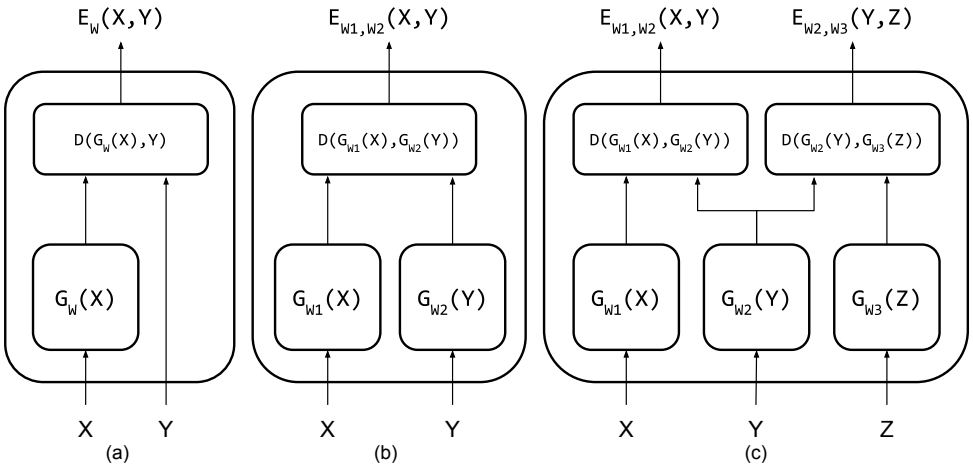

Figure 5.3 – Three EBM architectures. When all Gs and Ws are equal, (b) and (c) correspond to the siamese/triplet cases.

$$\mathcal{L} = \frac{1}{P}\sum_{i=1}^{P} E_W(X^i, Y^i) = \frac{1}{P}\sum_{i=1}^{P} \|G_W(X^i) - Y^i\|^2 \tag{5.2}$$

5.3.2 *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 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 Figure 5.3b), 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:

$$E_{W_1,W_2}(X, Y) = D(G_{W_1}(X), G_{W_2}(Y)) \tag{5.3}$$

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, Hadsell, and LeCun, 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 $\{C^i\}_{i=1}^P$. E.g., $C^i = 0$ 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 various 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:

$$E_W(X, Y) = 1 - \exp(-\|G_W(X) - G_W(Y)\|_1) \tag{5.4}$$

$$\mathcal{L} = \frac{1}{P}\sum_{i=1}^{P} \|E_W(X^i, Y^i) - C^i\|^2 \tag{5.5}$$

5.3.3 *Triplet architecture*

The triplet architecture (Wang et al., 2014; Hoffer and Ailon, 2015; Schroff, Kalenichenko, and Philbin, 2015), as can be seen in Figure 5.3c, 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:

$$\mathcal{L} = \frac{1}{2P}\sum_{i=1}^{P} (\|ne^+ - 0\|^2 + \|ne^- - 1\|^2) \tag{5.6}$$

$$ne^+ = \frac{e^{E_W(X^i, Y^i)}}{e^{E_W(X^i, Y^i)} + e^{E_W(Y^i, Z^i)}} \tag{5.7}$$

$$ne^- = \frac{e^{E_W(Y^i, Z^i)}}{e^{E_W(X^i, Y^i)} + e^{E_W(Y^i, Z^i)}} \tag{5.8}$$

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 the next subsection.

5.3.4 *On our choice of loss functionals*

The softmax triplet loss (STL) performed better in our experiments than the margin-based triplet loss used in Schroff, Kalenichenko, and Philbin (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 Figure 5.4a). In other words, the distance between the positive and the negative is at least m (when all three points are aligned).

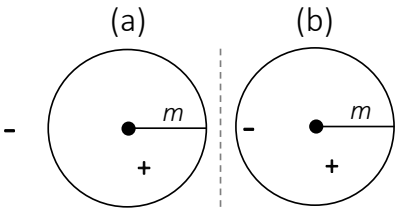

Figure 5.4 – •, -, + denote anchor, negative, and positive.

However, the objective of STL is simply $d^- > d^+$, without imposing an absolute lower bound on the distance between positives and negatives (i.e., only the distance ratio is of interest, see Figure 5.4b), which gives more freedom to the model.

For consistency, we also adopt a margin-free loss functional for siamese (MSE, see Equation 5.5). It also performed better than the traditional contrastive loss (Chopra, Hadsell, and LeCun, 2005; Neculoiu, Versteegh, and Rotaru, 2016) in early experiments.

5.3.5 *Sampling procedures*

We sample tuples from the ground truth abstractive communities to train our utterance encoder G_W (see Section 5.5) 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. As explained in the previous subsection, the softmax triplet loss captures a finer notion of distance than margin-based losses. This allows us to propose a novel, flexible *triplet sampling scheme* that enables subtle patterns, such as overlapping and nested groupings (e.g., human annotated abstractive communities, see Figure 5.2), to be learned. Full details are provided in the next section.

5.4 PROPOSED TRIPLET SAMPLING SCHEME

Recall that for a given triplet $(\text{pos}, \text{anc}, \text{neg})$ (positive, anchor, negative), the objective of training is to make the distance between pos and anc much smaller than the distance between anc and neg. We construct triplets by considering community pairs. For a meeting that includes N unique communities, we have $\binom{N}{2}$ unique pairs of them. A given pair

of communities can either (1) be disjoint, i.e., not have any element in common (Figure 5.5a), (2) be nested in one another (top of Figure 5.5b), or (3) overlap (Figure 5.5c). Our triplet sampling scheme needs to account for these three cases so that the learned embedding space encodes a meaningful distance that can recover such fine patterns.

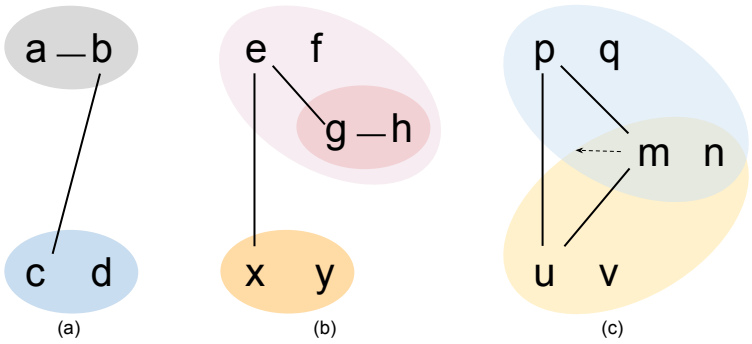

Figure 5.5 – (a) communities $\{a, b\}$ and $\{c, d\}$ are disjoint (b) community $\{g, h\}$ is nested in community $\{e, f, g, h\}$ (c) communities $\{p, q, m, n\}$ and $\{m, n, u, v\}$ overlap upon $\{m, n\}$.

5.4.1 *Disjoint case*

As shown in Figure 5.5a, let us consider two communities $\{a, b\}$ and $\{c, d\}$ that do not share any element. We can derive 8 triplets from these two communities:
$(a, b, c), (b, a, c), (a, b, d), (b, a, d), (c, d, a), (d, c, a), (c, d, b), (d, c, b)$.

As a result of passing these triplets to the network, the intra-community distances are reduced while the inter-community distances are enlarged. Note that (a, b, c) and (b, a, c) are considered different triplets, as they involve different sides of the same triangle.

To formalize, we denote the set of all 2-permutations of a set S by $\text{Permu}(S, 2) = \{(i, j) : i, j \in S, i \neq j\}$, and the number of such permutations as $P_2^{|S|} = \frac{(|S|)!}{(|S|-2)!}$, where $|*|$ denotes cardinality. Note that when $|S| = 1$, (singleton community), we repeat the single element of $S = \{i\}$, thus $\text{Permu}(S, 2) = \{(i, i)\}$. The Cartesian product of two sets $S1$ and $S2$ is denoted as $\text{Carte}(S1, S2) = \{(i, j) : i \in S1, j \in S2\}$, the universal set of all elements as Ω (union of all communities), and the empty set as $\emptyset$.

For any two disjoint communities, i.e., $A \cap B = \emptyset$, the complete set of triplets we can sample from is:

$$\begin{aligned} &\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Permu}(A, 2), \text{neg} \in B\} \\ \cup &\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Permu}(B, 2), \text{neg} \in A\} \end{aligned} \tag{5.9}$$

The corresponding number of triplets is $P_2^{|A|} \times |B| + P_2^{|B|} \times |A|$.

5.4.2 *Nested case*

Some communities are nested. E.g., as shown in Figure 5.5b, community $\{g,h\}$ is nested in community $\{e,f,g,h\}$. In that case, we first create triplets by comparing the nested part $\{g,h\}$ with the difference between the larger community and the smaller one, i.e., with $\{e,f,g,h\} \setminus \{g,h\} = \{e,f\}$. These two parts are disjoint, so we can use the sampling scheme previously described (Subsection 5.4.1). This ensures, e.g., that $d_{hg} \ll d_{ge}, d_{ef} \ll d_{fh}$, etc.

Second, we should apply an extra constraint to the edges linking $\{g,h\}$ and $\{e,f\}$, such as d_{ge}, in order to guarantee that g is closer to e than to any object from any other disjoint community (e.g., x or y). That is, we want $d_{ge} \ll d_{ex}$. Therefore, we create the following extra triplets:
$(e,g,x), (g,e,x), (e,g,y), (g,e,y), (e,h,x), (h,e,x), (e,h,y), (h,e,y)$
$(f,g,x), (g,f,x), (f,g,y), (g,f,y), (f,h,x), (h,f,x), (f,h,y), (h,f,y)$.

To formalize, for any two nested communities, e.g. $A \subset B$, the complete set of triplets we can sample from is:

$$\begin{aligned} &\{(\text{pos},\text{anc},\text{neg}) : (\text{pos},\text{anc}) \in \text{Carte}(A, B \setminus A), \text{neg} \in \Omega \setminus B\} \\ \cup &\{(\text{pos},\text{anc},\text{neg}) : (\text{pos},\text{anc}) \in \text{Carte}(B \setminus A, A), \text{neg} \in \Omega \setminus B\} \\ \cup &\{(\text{pos},\text{anc},\text{neg}) : (\text{pos},\text{anc}) \in \text{Permu}(A, 2), \text{neg} \in B \setminus A\} \\ \cup &\{(\text{pos},\text{anc},\text{neg}) : (\text{pos},\text{anc}) \in \text{Permu}(B \setminus A, 2), \text{neg} \in A\} \end{aligned} \tag{5.10}$$

The corresponding number of triplets is $|A| \times |B \setminus A| \times |\Omega \setminus B| \times 2 + P_2^{|A|} \times |B \setminus A| + P_2^{|B \setminus A|} \times |A|$.

5.4.3 *Overlapping case*

As shown in Figure 5.5c, two communities may overlap. E.g., $\{p,q,m,n\}$ and $\{m,n,u,v\}$ overlap upon $\{m,n\}$. We first consider this overlapping case as two nested cases: $\{m,n\}$ nested in $\{p,q,m,n\}$ and $\{m,n\}$ nested in $\{m,n,u,v\}$. We thus sample triplets as explained in Subsection 5.4.2.

However, here, we also have the extra constraint that the overlap $\{m,n\}$ be pulled in-between $\{p,q\}$ and $\{u,v\}$, as shown by the dashed arrow in Figure 5.5c. Note that here, in order to teach the model that $\{m,n\}$ should be placed between $\{p,q\}$ and $\{u,v\}$, m and n can only be used as positives. Indeed, if they were used as anchor or negatives, the model would only learn to push them away from other objects, (respectively, from the negative and anchor). Thus, we sample the following additional triplets:
$(m,p,u), (m,u,p), (m,p,v), (m,v,p), (m,q,u), (m,u,q), (m,q,v), (m,v,q)$
$(n,p,u), (n,u,p), (n,p,v), (n,v,p), (n,q,u), (n,u,q), (n,q,v), (n,v,q)$.
For instance, adding triplets (m,p,u) and (m,u,p) imposes simultaneously $d_{mp} \ll d_{pu}$ and $d_{mu} \ll d_{up}$, a constraint best fulfilled when m is placed in the middle of p and u.

To formalize, for any two overlapping communities A and B, i.e., $A \cap B \neq \emptyset, A \neq B, A \not\subset B, B \not\subset A$, the additional triplets correspond to:

$$\begin{aligned}
&\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Carte}(A \cap B, A \setminus B), \text{neg} \in B \setminus A\} \\
&\cup\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Carte}(A \cap B, B \setminus A), \text{neg} \in A \setminus B\} \\
&\hline \\
&\cup\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Permu}(A \setminus B, 2), \text{neg} \in B \setminus A\} \\
&\cup\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Permu}(B \setminus A, 2), \text{neg} \in A \setminus B\}
\end{aligned}$$

And, as was previously mentioned, we also consider two nested cases:

$$\begin{aligned}
&\cup\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Carte}(A \setminus B, A \cap B), \text{neg} \in \Omega \setminus (A \cup B)\} \\
&\cup\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Carte}(A \cap B, A \setminus B), \text{neg} \in \Omega \setminus (A \cup B)\} \\
&\cup\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Permu}(A \setminus B, 2), \text{neg} \in A \cap B\} \\
&\cup\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Permu}(A \cap B, 2), \text{neg} \in A \setminus B\} \\
&\hline \\
&\cup\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Carte}(B \setminus A, A \cap B), \text{neg} \in \Omega \setminus (A \cup B)\} \\
&\cup\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Carte}(A \cap B, B \setminus A), \text{neg} \in \Omega \setminus (A \cup B)\} \\
&\cup\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Permu}(B \setminus A, 2), \text{neg} \in A \cap B\} \\
&\cup\{(\text{pos}, \text{anc}, \text{neg}) : (\text{pos}, \text{anc}) \in \text{Permu}(A \cap B, 2), \text{neg} \in B \setminus A\}
\end{aligned} \tag{5.11}$$

The corresponding total number of triplets is therefore:

$$\begin{aligned}
&|A \cap B| \times |A \setminus B| \times |B \setminus A| \times 2 \\
&+P_2^{|A \setminus B|} \times |B \setminus A| + P_2^{|B \setminus A|} \times |A \setminus B| \\
&+|A \setminus B| \times |A \cap B| \times |\Omega \setminus (A \cup B)| \times 2 + P_2^{|A \setminus B|} \times |A \cap B| + P_2^{|A \cap B|} \times |A \setminus B| \\
&+|B \setminus A| \times |A \cap B| \times |\Omega \setminus (A \cup B)| \times 2 + P_2^{|B \setminus A|} \times |A \cap B| + P_2^{|A \cap B|} \times |B \setminus A|
\end{aligned} \tag{5.12}$$

5.4.4 *Visualization*

Our triplet sampling scheme is effective if it can make the triplet architecture learn an embedding space in which distances capture basic community structure (disjoint case) and the two more subtle nested and overlapping cases previously tackled. In order to test its effectiveness, we inspect what happens for the IS1001c meeting, which includes 12 abstractive communities and 48 unique utterances. For this meeting, 23612 triplets can be sampled with our approach.

We trained our model (see Section 5.5) on this set of triplets for 5 epochs. The utterance embeddings projected onto the first two PCA

dimensions are shown in Figure 5.6, in which the utterances belonging to the same ground truth community are encircled by an ellipse (marked with a letter in $\{A, \dots, L\}$).

We can observe that utterances are placed at the right places as desired. Overall, utterances belonging to the same community are close to each other (small intra-community distances), and disjoint communities (e.g., A, B, and C) are far from each other (large inter-community distances). In the upper corner of Figure 5.6, the nested community J and the part I \ J are well separated, but they are closer to each other than to any other disjoint community. In the center of Figure 5.6, the overlap K ∩ L has successfully been pulled in-between the two overlapping communities.

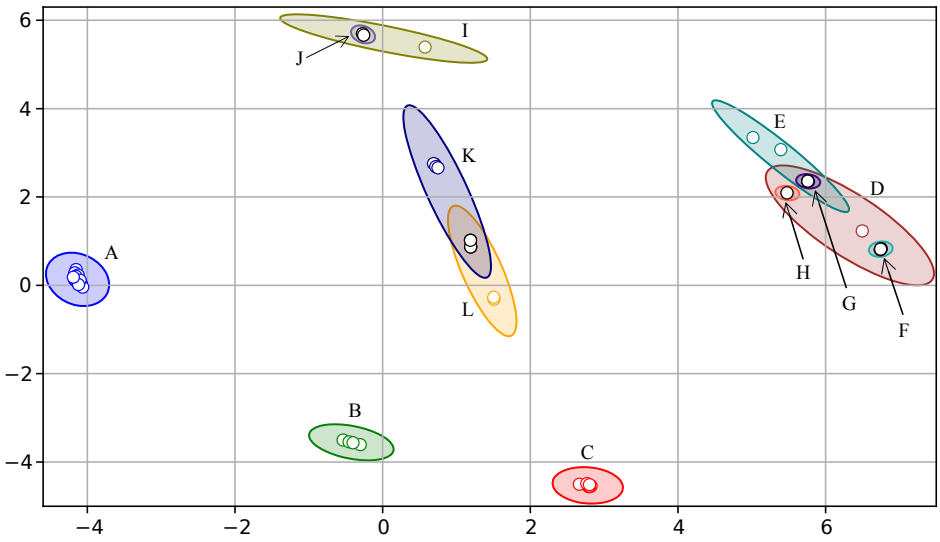

Figure 5.6 – All 48 utterances of 12 abstractive communities from the meeting IS1001c projected into 2-dimensional PCA of learned 32-dimensional embedding space. Trained on 23612 triplets for 5 epochs. Converged $P@k = v$ is equal to 96.33%.

This provides evidence that, with well-designed triplets, it is possible to learn a space encoding very fine clustering patterns with the triplet architecture. To the best of our knowledge, this is novel and has never been proven by prior literature. Moreover, our proposed triplet sampling scheme is general and can thus be applied to any task where a rich metric needs to be learned to encode subtle grouping information.

The embedding space can also be interactively explored. We provide below a link to Google's Embedding Projector, corresponding to a more complicated example with more elements and communities (the ES2016b meeting) learned in the same way as in the previous example:

https://projector.tensorflow.org/?config=https://gist.githubusercontent.com/shangguokan/fb859f90563369cc6b01e7897ec6fb37/raw/063f2429f46cee13896a66a9aae059934aef16a2/ES2016b.json

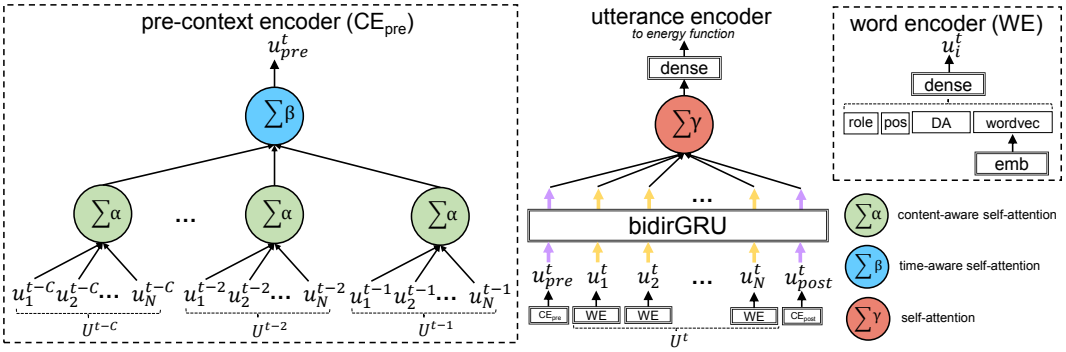

Figure 5.7 – Our proposed utterance encoder. Only the pre-context encoder is shown. C is the context size.

5.5 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. Upper and lower case are used for matrices and vectors. Vectors are distinguished from floats by using boldface.

5.5.1 *Word encoder*

As shown in the upper right corner of Figure 5.7, 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.

5.5.2 *Utterance encoder*

As shown in the center of Figure 5.7, 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 Figure 5.2), 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 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. How we derive the pre and post-context vectors is explained in Subsection 5.5.3.

Self-attention. The self-attention mechanism (Yang et al., 2016b; Lin et al., 2017; Vaswani et al., 2017), also called *inner* or *intra* attention, emerged in the literature following the success of attention in the sequence-to-sequence setting (Bahdanau, Cho, and Bengio, 2015; Luong, Pham, and Manning, 2015). While self-attention deals with a single source sequence (no decoder), the motivation is the same as with traditional attention: rather than considering the last annotation of the RNN encoder as a summary of the entire input sequence, which is prone to information loss, a new hidden representation is computed as a weighted sum of the annotations at *all* positions, where the weights are computed by a trainable mechanism that performs a comparison operation.

While in seq2seq, the comparison involves the transformed input and the current hidden state of the decoder, in the encoder-only setting, the annotations $\mathbf{H}^t$ are passed through a dense layer and compared (dot product) with a trainable vector $\mathbf{u}_\gamma$, initialized randomly. Then, a probability distribution over the $N+2$ tokens in $\mathbf{U}^t$ is obtained via a softmax:

$$\boldsymbol{\gamma}^t = \text{softmax}(\mathbf{u}_\gamma \cdot \tanh(\mathbf{W}_\gamma \mathbf{H}^t)) \tag{5.13}$$

(bias omitted for readability). The attentional vector for $\mathbf{U}^t$ is finally computed as a weighted sum of its annotations, and, as shown in Figure 5.7, is finally passed to a dense layer to obtain the utterance embedding $\mathbf{u}^t \in \mathbb{R}^{d_f}$:

$$\mathbf{u}^t = \text{dense}\left(\sum_{i=1}^{N+2} \gamma_i^t \mathbf{h}_i^t \right) \tag{5.14}$$

$\mathbf{u}_\gamma$ replaces the hidden state of the decoder in the traditional attention mechanism. It 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.

5.5.3 *Context encoder: level 1*

We now explain how we derive the pre and post-context vectors that we prepend and append to $\mathbf{U}^t$ so as to inject contextual information into the encoding process. They are obtained by aggregating information from the C utterances preceding and following $\mathbf{U}^t$ (respectively):

$$\mathbf{u}^t_{\text{pre}} \leftarrow \text{aggregate}_{\text{pre}}(\{\mathbf{U}^{t-C}, \dots, \mathbf{U}^{t-1}\}) \tag{5.15}$$

$$\mathbf{u}^t_{\text{post}} \leftarrow \text{aggregate}_{\text{post}}(\{\mathbf{U}^{t+1}, \dots, \mathbf{U}^{t+C}\}) \tag{5.16}$$

where C, the context size, is a hyperparameter. Since $\mathbf{u}^t_{\text{pre}}$ and $\mathbf{u}^t_{\text{post}}$ 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 Figure 5.7). The pre and post-context encoders share the exact 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 $\mathbf{U}^{t-1}$:

$$\boldsymbol{\alpha}^{t-1} = \text{softmax}\Big(\mathbf{u}_{\square} \cdot \tanh\big(\mathbf{W}_{\square}\mathbf{U}^{t-1} + \mathbf{W}' \sum_{i=1}^{N} \mathbf{u}_i^t\big)\Big) \tag{5.17}$$

This mechanism is the same as in Equation 5.13, 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, Liu, and Manning, 2017b). In See, Liu, and Manning (2017b), the coverage vector is the sum, over all previous steps of the decoder, of the attentional distributions over the source words. Its role is to decrease repetition in the final summary, by letting the attention mechanism know which information about the source document has already been captured, in the hope that the model will focus on other aspects of it. 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.

Bi-directional information pathway. To recapitulate, we consider $\mathbf{U}^t$ when computing $\mathbf{u}^t_{\text{pre}}$ and $\mathbf{u}^t_{\text{post}}$, and then prepend/append these vectors to $\mathbf{U}^t$ when encoding it. Therefore, in effect, information first flows from the current utterance to its context to guide context encoding, and then flows back to the current utterance encoding mechanism.

Weight sharing. The same content-aware self-attention mechanism is applied to the entire context surrounding $\mathbf{U}^t$, that is, to all preceding and following utterances. We did experiment with separate pre/post mechanisms, without significant improvements. This makes sense, as there is no inherent difference between preceding and following utterances. Indeed, the latter become the former as we slide the window over the transcription from start to finish. In addition, sharing weights makes for 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).

Dimensionality reduction. 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 $\mathbf{u}^t_{\text{pre}} \in \mathbb{R}^d$ as described next.

5.5.4 *Context encoder: level 2*

As can be seen in Figure 5.2, 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 used the trainable universal time-decay attention mechanism of Su, Yuan, and Chen (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:

$$\beta^{t-1} = w_1 \beta^{\text{conv}^{t-1}} + w_2 \beta^{\text{lin}^{t-1}} + w_3 \beta^{\text{conc}^{t-1}} \tag{5.18}$$

$$= \frac{w_1}{a(d^{t-1})^b} + w_2[ed^{t-1} + k]^+ + \frac{w_3}{1 + (\frac{d^{t-1}}{D_0})^l} \tag{5.19}$$

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 each model a different type of time dependence. Respectively, they 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.

To sum up, across the utterance and the context encoders, our architecture makes use of three different attention mechanisms.

5.6 COMMUNITY DETECTION

Once the utterance encoder G_W presented in Section 5.5 has been trained within the siamese meta-architecture or triplet meta-architecture (with the triplet sampling scheme of Section 5.4) presented in Section 5.3, 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, Ehrlich, and Full, 1984).

FCM returns a probability distribution over all communities for each utterance.

Fuzzy c-Means algorithm. More specifically, the goal of the FCM algorithm is to minimize the weighted within group sum of squared error objective function:

$$J(M,Q) = \sum_{q=1}^{|Q|}\sum_{t=1}^{T}(m_{qt})^{fuz}\|\mathbf{u}^t - \mathbf{c}_q\|_2^2 \tag{5.20}$$

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, $\|.\|_2$ denotes the Euclidean distance in the triplet case (we replace it with Manhattan distance $\|.\|_1$ 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:

$$m_{qt} = \Big(\sum_{j=1}^{|Q|}\big(\frac{\|\mathbf{u}^t - \mathbf{c}_q\|_2}{\|\mathbf{u}^t - \mathbf{c}_j\|_2}\big)^{\frac{2}{fuz-1}}\Big)^{-1} \tag{5.21}$$

$$\mathbf{c}_q = \frac{\sum_{t=1}^{T}(m_{qt})^{fuz}\mathbf{u}^t}{\sum_{t=1}^{T}(m_{qt})^{fuz}} \tag{5.22}$$

When $fuz \to +\infty$, $\forall q \in |Q|$, $\forall t \in T$, m_{qt} tends to be equal to $1/|Q|$, thus utterances have identical membership to each community. While when $fuz \to 1$, 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 until the maximum number of iterations is reached or $J(M,Q)$ decreases by less than a predefined threshold.

5.7 EXPERIMENTAL SETUP

5.7.1 *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 Figure 5.2. There are 2368 unique abstractive communities in total, whose statistics are shown in Table 5.1. We adopt the officially suggested *scenario-only partition* [3], which provides 97, 20, and

3. http://groups.inf.ed.ac.uk/ami/corpus/datasets.shtml

type	abstract	action	problem	decision	total
unique	1147	247	380	594	2368
disjoint	528	124	69	45	766
nested	96	106	200	437	839
overlapping	349	17	163	149	678
singleton	49	162	38	244	493

Table 5.1 – Statistics of abstractive communities.

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`.

5.7.2 *Baselines*

First, we evaluate our utterance encoder against two *baseline encoders* LD and HAN that are trained within the energy framework, as well as 4 variants of our model. Note that to be fair, we ensure that both LD and HAN have access to context. We then compare our full pipeline against unsupervised and supervised *baseline systems*. Full details are provided in below.

Baseline encoders

— **LD** (Lee and Dernoncourt, 2016) is a sequential sentence encoder developed for dialogue act classification. The model takes into account a fixed number of utterances only 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 Dernoncourt (2016) reported them to work better than GRU cells in their experiments.

— **HAN** (Yang et al., 2016b). 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 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 5.8.

Variants of our model

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.

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 5.7.3). 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, Mitra, and Sugiyama, 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 ($|Q| = v/11$, see Subsection 5.7.4), and then only summary-worthy utterances within segments are retained for evaluation.

Supervised baseline systems

As discussed in the literature review (see Section 5.2), original approaches to ACD (Murray, Carenini, and Ng, 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 (Girvan and Newman, 2002), is applied to detect overlapping communities on the utterance graph.

5.7.3 *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, Le, and Sutskever (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 5.8.

Optimization. The exact same token representations and settings were used for our model, its variants, and the baseline encoders. 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.

Tuple subsampling and resampling. When labeled tuples (pairs or triplets) are not provided, but must be constructed from the dataset, subsampling is critical for training a model within the siamese or triplet meta-architectures. For instance, in face verification (Chopra, Hadsell, and LeCun, 2005), a virtually infinite number of impostor pairs can be constructed, while only a limited number of genuine pairs are available. Usually, one selects $n \geq 1$ times more impostor pairs than genuine pairs, but n must not be too large to avoid large imbalance (Chopra, Hadsell, and LeCun, 2005; Neculoiu, Versteegh, and Rotaru, 2016). Subsampling is also a critical issue for triplets (Wang et al., 2014; Schroff, Kalenichenko, and Philbin, 2015; Amos, Ludwiczuk, and Satyanarayanan, 2016).

Following (Hoffer and Ailon, 2015; Liu et al., 2019), at the beginning of each epoch, we sample one triplet from each pair of communities belonging to the same meeting, using the strategy explained in Section 5.4. We thus obtain 15594 training 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.

While the training tuples were resampled at each epoch, we used a fixed validation set of 2891/5782 triplets/pairs. On the test set, no (re)sampling is necessary: we simply get a vector for each utterance by feeding them to the trained model G_W.

5.7.4 *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.

Distance level

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 is provided in Subsection 5.9.2.

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 5.3.2 and 5.3.3).

Clustering level

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, Carenini, and Ng, 2012). A detailed explanation is provided in Subsection 2.2.3.

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 $|Q|$ ($|Q| = v$), equal to the number of ground truth communities.

Note that since FCM does not return nested groupings, we merged the ground truth communities nested under the same community. Moreover, due to its stochastic nature, we run the algorithm 20 times with different random initializations and select the run yielding the smallest objective function value.

5.8 QUANTITATIVE 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: $(\text{pre}, \text{post}) = (C, 0)$, and $(\text{pre}, \text{post}) = (C, C)$. Results are shown in Figure 5.8. 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.

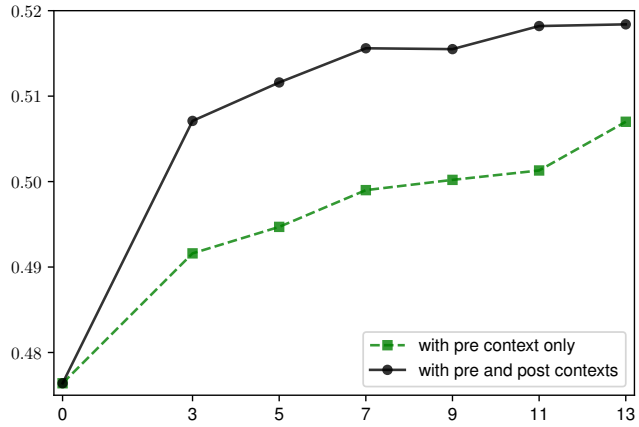

Figure 5.8 – Impact of context size on the validation $P@k = v$, for our model trained within the triplet meta-architecture.

Performance comparison. Final test set results are shown in Table 5.2. 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 unsupervised/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,n1,01) or short context (k2,l2,n2,02), but not with large contexts (k3,l3,n3,03). Results also show that overall, using larger contexts always brings improvement.

			(pre, post)	P @$k = v$	P @$k = 10$	R @$k = 10$	F1 @$k = 10$	Omega index ×100 $\lvert Q\rvert = v$	Omega index ×100 $\lvert Q\rvert = 11$
	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
Triplet	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
	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***
Siamese	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
	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
Unsupervised	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**
	n1)	tf-idf	(0, 0)	-	-	-	-	25.04	25.14
	n2)	tf-idf	(3, 0)	-	-	-	-	27.33	26.95
Supervised	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 5.2 – 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.

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 5.1). For this simplified task, we have 1147 unique 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 $|Q| = 8$.

Usage of discourse features. Discourse features are very helpful through our experiments. They are introduced into our model by concatenating at the word-level (see Section 5.5.1) instead of concatenating with the output of self-attention γ at the sentence-level. The decision is made based on empirical results, moreover this is also aligned with the nature of word being part of transcription, which has richer meaning than just the word itself, thus its representation should be enriched with discourse features.

5.9 QUALITATIVE RESULTS

In this section, we first visualize that the three self-attention mechanisms behave in a cooperative manner to produce a meaningful utterance representation. We also visualize the attention coefficients of the two time-aware self-attention mechanisms, and find that interestingly, the distributions over the pre and post-context are not symmetric. We then inspect the closest utterances to a given query utterance.

5.9.1 *Attention visualization*

The aim of this subsection is to show, with an example, what the three self-attention mechanisms pay attention to while encoding the current utterance $\mathbf{U}^t$ (here, an utterance from the ES2011c *validation* meeting). Figure 5.9 shows the attention distributions over $\mathbf{U}^t$ (highlighted by the black frame), and over its pre-context $\{\mathbf{U}^{t-1}, \dots, \mathbf{U}^{t-11}\}$ and post-context $\{\mathbf{U}^{t+1}, \dots, \mathbf{U}^{t+11}\}$ utterances. We use three colors that are consistent with the ones used in Figure 5.7 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 Figure 5.9 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

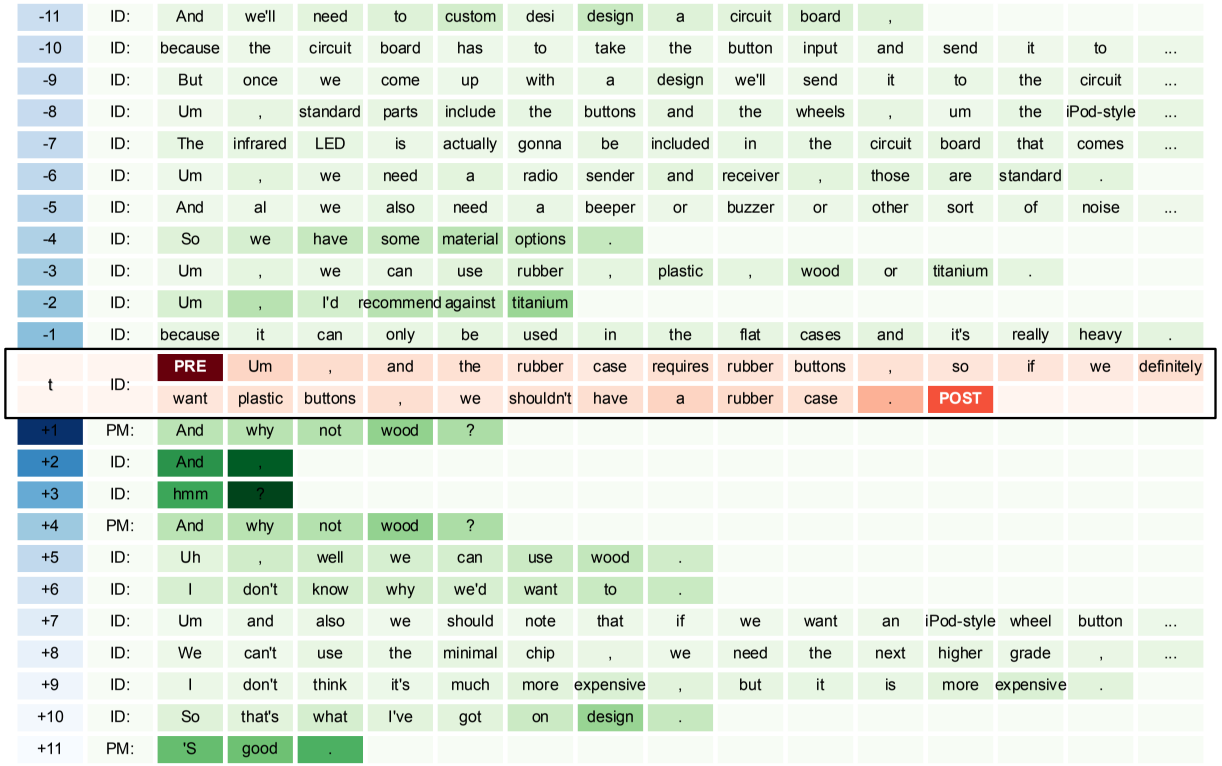

Figure 5.9 – Visualization of attention distributions around an utterance from the ES2011c meeting. Some utterances are truncated for readability.

as: "custom", "design" from $\mathbf{U}^{t-11}$, "material" from $\mathbf{U}^{t-4}$, "recommend", "titanium" from $\mathbf{U}^{t-2}$, "wood" from $\mathbf{U}^{t+1}$, etc.

— The time-aware self-attention mechanism β (blue) places more importance over the context utterances that are close to $\mathbf{U}^t$, i.e., the importance decreases when the time distance increases. However, the patterns are different for the pre and post-contexts (see Figure 5.10).

— 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 $\{\mathbf{U}^{t-11}, \dots, \mathbf{U}^{t}, \dots, \mathbf{U}^{t+11}\}$. 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.

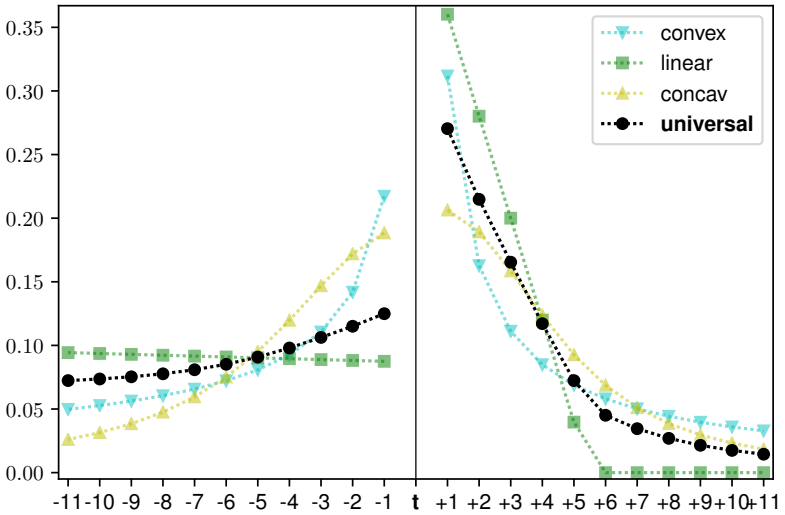

Figure 5.10 – Normalized time-aware self-attention weights for pre and post-contexts, averaged over 10 runs.

We also inspect in Figure 5.10 the attention coefficients of the time-aware self-attention mechanisms (see Equation 5.18) equipping the

pre and post-context encoders. It is interesting to observe that the distributions are not symmetric. Indeed, only the utterances immediately following $\mathbf{U}^t$ $(t+1 \rightarrow t+5)$ seem to matter, while the attention weights are much more uniform across the utterances preceding $\mathbf{U}^t$. This suggests that in dialogues, considering a long history of preceding utterances helps understanding the current one.
It is also interesting to note that the parameters that have been learned for the pre-context linear function make it increasing, rather than decreasing. This is counter-intuitive, but allowed by design. Overall though, the three terms altogether do produce a function that slowly decreases as time distance increases, which is in accordance with intuition.

5.9.2 *Ranking example*

For the same utterance from the ES2011c meeting as used in Subsection 5.9.1, we show in Table 5.3 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 2^{nd} and 7^{th}, 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 Table 5.3) 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`.

5.10 CONCLUSION AND FUTURE WORK

This work 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. Results also show that energy-based modeling is well-suited to ACD. Moreover, we show that

dist	pos	DA	role	text
0	*t*	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 5.3 – Ranking example.

our general triplet sampling scheme enables the triplet architecture to learn subtle clustering patterns, such as overlapping and nested communities. This has many applications outside of ACD.

Future work should focus on (1) summarizing each community with an abstractive sentence (subtask `b` in Figure 5.1). New datasets such as `Dial2Desc` (Pan et al., 2018) and `SAMSum` (Gliwa et al., 2019) for short dialogue summarization , might be a remedy to the lack of a large-scale summarization dataset for the conversation domain; (2) predicting community types, i.e., `ABSTRACT`, `ACTIONS`, `PROBLEMS`, and `DECISIONS`, which could be useful for summarization; (3) applying our contextual utterance encoder to other tasks; (4) evaluating our triplet sampling scheme, along with tackling overlapping and nested clustering or multi-label classification tasks, on other datasets. In order to show the general applicability of the scheme, we plan to conduct experiments on graph, image and text synthetic/real-world datasets, and hopefully obtained results may shed some light in related fields.

6 CONCLUDING REMARKS

In this chapter, we conclude the dissertation by first summarizing our main contributions reported in detail in the previous chapters, and we then outline the promising future research directions unexplored at the time of writing.

6.1 SUMMARY OF CONTRIBUTIONS

In the context of the LinTO meeting assistant project, this dissertation has focused on teaching machines to understand multi-party meeting speech, and more specifically, to automatically generate meeting summaries. Our contributions have been made along with developing novel approaches to address three specific NLP/SLU tasks centered around the main subject of this thesis:

ABSTRACTIVE MEETING SUMMARIZATION, which aims to take a meeting speech transcription as input and generate an abstractive summary consisting of novel sentences. In Chapter 3 we introduced a novel approach to this task, which is fully unsupervised, does not rely on any particular annotations, and can be applied to any language in an almost out-of-the-box fashion. More specifically, we presented a graph-based NLG component MSCG, in which 1) a word graph is constructed based on a set of topically coherent utterances, 2) then graph edges and paths are weighted and re-ranked according to novel scoring functions developed by leveraging advances in word embeddings, graph-of-words, and graph degeneracy, and 3) finally, the best path is retrieved from the graph as a novel generated abstractive sentence. To form the final summary of a certain size, we select the best elements from the set of abstractive sentences by maximizing a custom submodular quality function under a budget constraint. Experiments showed that our system improves on the state-of-the-art and generates reasonably grammatical abstractive summaries despite taking noisy utterances as input.

DIALOGUE ACT CLASSIFICATION, whose goal is to assign each utterance a dialogue act label to represent its communicative intention. In Chapter 4, we introduced a modified neural CRF layer that takes speaker information into account for this task. More specifically, we made the label transition matrix of the CRF to be conditioned on speaker-change, i.e., two matrices that encode different DA transition patterns for the "speaker unchanged" and "speaker changed" cases. Our modified CRF layer is general and

can be plugged on top of any deep learning component to form a DA classification model. In our experiments, we evaluated it within the BiLSTM-CRF architecture, results showed that our modified CRF layer outperforms the classical one, with very wide margins for some DA labels. Further, visualizations demonstrated that our CRF layer can learn meaningful, sophisticated, and speaker-change-aware transition patterns between DA label pairs in an end-to-end way. The empirical results of this work confirmed our hypothesis that speaker information is indeed beneficial to the task of DA classification.

ABSTRACTIVE COMMUNITY DETECTION, in which utterances in a conversation are grouped according to whether they can be jointly summarized by a common abstractive sentence. In Chapter 5, we introduced an energy-based or deep metric learning approach to this task. More specifically, 1) we first presented a neural contextual utterance encoder featuring three types of self-attention mechanisms, which takes contextual and temporal information into account when embedding a target utterance. 2) We then trained it using the siamese and triplet architectures, whose objective is to project training utterances into an embedding space in which the utterances from a given abstractive community are close to each other. Here, we proposed a general sampling scheme that enables the triplet architecture to capture subtle clustering patterns, such as overlapping and nested communities. 3) Finally, we applied the Fuzzy c-Means clustering algorithm on the trained utterance embeddings in order to obtain abstractive communities. Experiments showed that our system outperforms multiple energy-based and non-energy based baselines from the state-of-the-art. Further, visualization results showed that the three self-attention mechanisms of our utterance encoder are able to adaptively focus on different information, in order to cooperatively produce a meaningful representation, and that our triplet sampling scheme is effective.

6.2 FUTURE WORK

We firmly believe that in order to enable unsupervised/supervised meeting summarization techniques to go one step further in performance, the systems will need to have a deeper understating of meetings and natural language.

DEEPER UNDERSTANDING OF MEETINGS. Current abstractive meeting summarization approaches mainly aim to produce *general or topic summaries*: long paragraphs that give an overview of what a conversation is about. However, they are often too general to be useful especially for business meetings, where *detailed or focused summaries* (anal-

ogous to human meeting minutes) are usually desired. Participants might want a record of the arguments made for or against a claim and who made them, for example, or to know the order in which it was decided that certain actions would be performed and who is meant to perform them. Generating such detailed summaries requires a thorough understanding of meeting structures, which is currently beyond the capabilities of state of the art summarization systems.

To accomplish this, we believe the systems need to exploit information about *rhetorical relations* that hold between the contents of dialogue acts in a conversation, together with the complex *discourse structures* that they entail (Thompson and Mann, 1987; Asher, 1993; Asher et al., 2003). For instance, instead of assuming that a conversation has a flat chain structure (a linear sequence of utterances) as we do nowadays, we can represent it as a discourse graph—a directed weakly connected graph reflecting the discourse structure—in which the nodes represent utterances and the edges represent discourse relations (e.g., elaboration, clarification, completion). In a nutshell, we would like to have a graph that encodes all aspects of the conversation, from low-level to higher-level understanding. Providing such graph as input to graph neural networks (Zhou et al., 2018) will allow a model to fully explore the information carried by the nodes (utterances) and edges (relations) and result in better abstractive summaries. A similar idea to what we propose here has been explored and proven promising for extractive summarization in the recent work of Xu et al. (2019), but as of yet, there has been no attempt to apply it to abstractive meeting summarization.

Apart from the discourse graph and going beyond the field of NLP, we can take advantage of multi-modal sensing of the meeting environment when building meeting summarization systems, such as microphones to capture speech and cameras to capture the individual participants and their interactions. Leveraging advances in audio and video processing to incorporate multi-modal information, such as head poses, eye gazes, facial expressions, hand gestures, and vocal emphasises, may improve performance over a system that draws on textual transcriptions alone. The recent work of Li et al. (2019a) has shown that the visual focus of attention, which is estimated based on each participant's head orientation and eye gaze, can assist their multi-modal summarization system in determining salient utterances and, consequently, improve the quality of generated abstractive meeting summaries. The assumption is that an utterance is more important if its speaker receives more attention from other meeting participants. We believe multi-modal systems can be one of the promising future research directions for abstractive meeting summarization, yet this is little explored in the literature (Erol, Lee, and Hull, 2003; Li et al., 2019a).

DEEPER UNDERSTANDING OF NATURAL LANGUAGE. Not only for meeting summarization but also for all NLP tasks, using a better text representation technique often yields a better performance. Recent advances in language models pre-trained on a large unannotated text corpus (e.g., BERT; Devlin et al., 2018) have brought many revolutionary advantages. For example, these models can provide word embeddings with an awareness of the context in which the word is used, so that e.g., `"apple"` will have different representations in different texts. Moreover, the pre-trained models can be easily plugged in as input to subsequent neural components, and then fine-tuned for specific often-low-resource tasks in a manner of *transfer learning* (Ruder et al., 2019). The systems built upon these language models have shown state-of-the-art performance on many NLP tasks, including summarization for traditional documents (Liu and Lapata, 2019), however this has not yet been confirmed on abstractive meeting summarization.

Moreover, Gururangan et al. (2020) have shown that instead of directly using a language model pre-trained on a massive, heterogeneous, and broad-coverage corpus, it is helpful in performance gains to first tailor the model to the domain of a target task (domain-adaptive pre-training), and to second adapt the model to the task's unlabeled data (task-adaptive pre-training). In our case, given an off-the-shelf pre-trained language model (e.g., BERT), we can conduct the continued pre-training of the model (e.g., masked language modeling objective) on domain-specific unlabeled data, such as speech transcriptions (e.g., radio, TV, call center), and then on task-specific unlabeled data, such as the meeting transcriptions of the AMI and ICSI corpora. The language model obtained with the multi-phase adaptive pre-training strategy above can be used in a sequence-to-sequence architecture and will potentially offer large gains in task performance of abstractive meeting summarization.

6.3 EPILOGUE

Meetings are increasingly a ubiquitous part of people's lives; throughout this dissertation we have presented our contributions to teach machines to understand multi-party meeting speech with a special interest in automatically generating abstractive meeting summaries. While important advances have been made in this field, a number of unanswered questions and challenging problems remain. I sincerely hope that the work described in this dissertation will shed some new light on constructing future studies that ultimately lead to resolving the task of abstractive meeting summarization.

BIBLIOGRAPHY

Joos, Martin (1950). « Description of language design. » In: *The Journal of the Acoustical Society of America* 22.6, pp. 701–707 (cit. on p. 15).

Harris, Zellig S (1954). « Distributional structure. » In: *Word* 10.2-3, pp. 146–162 (cit. on p. 15).

Allen, Kent, Madeline M Berry, Fred U Luehrs Jr, and James W Perry (1955). « Machine literature searching VIII. Operational criteria for designing information retrieval systems. » In: *American Documentation (pre-1986)* 6.2, p. 93 (cit. on p. 17).

Bellman, R., Rand Corporation, and Karreman Mathematics Research Collection (1957). *Dynamic Programming*. Rand Corporation research study. Princeton University Press. ISBN: 9780691079516. URL: https://books.google.fr/books?id=wdtoPwAACAAJ (cit. on p. 12).

Firth, John R (1957). « A synopsis of linguistic theory, 1930-1955. » In: *Studies in linguistic analysis* (cit. on p. 15).

Luhn, Hans Peter (1957). « A statistical approach to mechanized encoding and searching of literary information. » In: *IBM Journal of research and development* 1.4, pp. 309–317 (cit. on p. 12).

Bellman, R. and Karreman Mathematics Research Collection (1961). *Adaptive Control Processes: A Guided Tour*. Princeton Legacy Library. Princeton University Press. ISBN: 9780691079011. URL: https://books.google.fr/books?id=POAmAAAAMAAJ (cit. on p. 12).

Rath, GJ, A Resnick, and TR Savage (1961). « The formation of abstracts by the selection of sentences. Part I. Sentence selection by men and machines. » In: *American Documentation* 12.2, pp. 139–141 (cit. on p. 5).

Austin, John Langshaw (1962). *How to do things with words*. Vol. 88. Oxford university press (cit. on p. 47).

Swets, John A (1963). « Information retrieval systems. » In: *Science* 141.3577, pp. 245–250 (cit. on p. 16).

Bellman, Richard (1966). « Dynamic programming. » In: *Science* 153.3731, pp. 34–37 (cit. on p. 53).

MacQueen, J. (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*. Berkeley, Calif.: University of California Press, pp. 281–297. URL: https://projecteuclid.org/euclid.bsmsp/1200512992 (cit. on p. 82).

Viterbi, Andrew (1967). « Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. » In: *IEEE transactions on Information Theory* 13.2, pp. 260–269 (cit. on pp. 50, 53).

Searle, John Rogers (1969). *Speech acts: An essay in the philosophy of language*. Vol. 626. Cambridge university press (cit. on p. 47).

Jones, Karen Sparck (1972). « A statistical interpretation of term specificity and its application in retrieval. » In: *Journal of documentation* (cit. on p. 12).

Sacks, Harvey, Emanuel A. Schegloff, and Gail Jefferson (1974). « A Simplest Systematics for the Organization of Turn-Taking for Conversation. » In: *Language* 50.4, pp. 696–735. ISSN: 00978507, 15350665. URL: http://www.jstor.org/stable/412243 (cit. on p. 49).

Van Rijsbergen, C.J. (1979). *Information Retrieval*. Butterworth Heinemann. URL: https://books.google.fr/books?id=a76ExwEACAAJ (cit. on p. 17).

Seidman, Stephen B (1983). « Network structure and minimum degree. » In: *Social networks* 5.3, pp. 269–287. DOI: 10.1016/0378-8733(83)90028-X (cit. on p. 26).

Bezdek, James C., Robert Ehrlich, and William Full (1984). « FCM: The fuzzy c-means clustering algorithm. » In: *Computers & Geosciences* 10.2, pp. 191 –203. ISSN: 0098-3004. DOI: 10.1016/0098-3004(84)90020-7. URL: http://www.sciencedirect.com/science/article/pii/0098300484900207 (cit. on p. 82).

Thompson, Sandra A and William C Mann (1987). « Rhetorical structure theory: A framework for the analysis of texts. » In: *IPRA Papers in Pragmatics* 1.1, pp. 79–105 (cit. on p. 99).

Collins, Linda M. and Clyde W. Dent (1988). « Omega: A General Formulation of the Rand Index of Cluster Recovery Suitable for Non-disjoint Solutions. » In: *Multivariate Behavioral Research* 23.2. PMID: 26764947, pp. 231–242. DOI: 10.1207/s15327906mbr2302_6. eprint: https://doi.org/10.1207/s15327906mbr2302_6. URL: https://doi.org/10.1207/s15327906mbr2302_6 (cit. on pp. 17, 88).

Salton, Gerard and Christopher Buckley (1988). « Term-weighting approaches in automatic text retrieval. » In: *Information processing & management* 24.5, pp. 513–523 (cit. on p. 12).

Asher, Nicholas (1993). *Reference to abstract objects in English* (cit. on p. 99).

Bromley, Jane, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah (1994). « Signature Verification using a "Siamese" Time Delay Neural Network. » In: *Advances in Neural Information Processing Systems 6*. Ed. by J. D. Cowan, G. Tesauro, and J. Alspector. Morgan-Kaufmann, pp. 737–744. URL: http://papers.nips.cc/paper/769-signature-verification-using-a-siamese-time-delay-neural-network.pdf (cit. on p. 72).

Meteer, Marie W, Ann A Taylor, Robert MacIntyre, and Rukmini Iyer (1995). *Dysfluency annotation stylebook for the switchboard corpus*. University of Pennsylvania Philadelphia, PA (cit. on p. 56).

Miller, George A. (Nov. 1995). « WordNet: A Lexical Database for English. » In: *Commun. ACM* 38.11, pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: http://doi.acm.org/10.1145/219717.219748 (cit. on p. 28).

Alexanderssony, Jan, Bianka Buschbeck-Wolfz, Tsutomu Fujinamiz, Elisabeth Maiery, Norbert Reithingery, Birte Schmitzx, and Melanie Siegelyy (1997). « Dialogue Acts in VERBMOBIL-2. » In: *DFKI* (cit. on p. 47).

Allen, James and Mark Core (1997). *Draft of DAMSL: Dialog act markup in several layers* (cit. on p. 47).

Core, Mark G and James Allen (1997). « Coding dialogs with the DAMSL annotation scheme. » In: *AAAI fall symposium on communicative action in humans and machines*. Vol. 56. Boston, MA, pp. 28–35 (cit. on p. 47).

Hochreiter, Sepp and Jürgen Schmidhuber (1997). « Long short-term memory. » In: *Neural computation* 9.8, pp. 1735–1780 (cit. on p. 16).

Jurafsky, Dan, Liz Shriberg, and Debra Biasca (1997). « Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual. » In: *Institute of Cognitive Science Technical Report*. URL: https://web.stanford.edu/~jurafsky/ws97/manual.august1.html (cit. on pp. 47, 54, 57).

Jones, K Sparck et al. (1999). « Automatic summarizing: factors and directions. » In: *Advances in automatic text summarization*, pp. 1–12 (cit. on p. 8).

Manning, Christopher and Hinrich Schutze (1999). *Foundations of statistical natural language processing*. MIT press (cit. on p. 11).

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd (1999). *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab (cit. on p. 36).

Ries, Klaus (1999). « HMM and neural network based speech act detection. » In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*. Vol. 1. IEEE, pp. 497–500 (cit. on p. 50).

Edmunds, Angela and Anne Morris (2000). « The problem of information overload in business organisations: a review of the literature. » In: *International journal of information management* 20.1, pp. 17–28 (cit. on p. 3).

Stolcke, Andreas, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer (2000). « Dialogue act modeling for automatic tagging and recognition of conversational speech. » In: *Computational Linguistics* 26.3, pp. 339–374. URL: https://www.aclweb.org/anthology/J00-3003 (cit. on pp. 47, 50, 51, 54).

Zechner, Klaus and Alex Waibel (2000). « Minimizing Word Error Rate in Textual Summaries of Spoken Language. » In: *1st Meeting of the North American Chapter of the Association for Computational Linguistics*. URL: https://www.aclweb.org/anthology/A00-2025 (cit. on p. 5).

Huang, Xuedong, Alex Acero, Hsiao-Wuen Hon, and Raj Reddy (2001). *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. 1st. USA: Prentice Hall PTR. ISBN: 0130226165 (cit. on p. 11).

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira (2001). « Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. » In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 282–289. ISBN: 1558607781 (cit. on p. 48).

Romano, Nicholas C and Jay F Nunamaker (2001). « Meeting analysis: Findings from research and practice. » In: *Proceedings of the 34th annual Hawaii international conference on system sciences*. IEEE, 13–pp (cit. on p. 1).

Batagelj, Vladimir and Matjaž Zaveršnik (2002). « Generalized cores. » In: *arXiv preprint cs/0202039* (cit. on p. 27).

Girvan, Michelle and Mark EJ Newman (2002). « Community structure in social and biological networks. » In: *Proceedings of the national academy of sciences* 99.12, pp. 7821–7826 (cit. on p. 86).

Zechner, Klaus (2002). « Automatic Summarization of Open-Domain Multiparty Dialogues in Diverse Genres. » In: *Computational Linguistics* 28.4, pp. 447–485. DOI: 10.1162/089120102762671945. URL: https://www.aclweb.org/anthology/J02-4003 (cit. on p. 4).

Asher, N., N.M. Asher, A. Lascarides, Cambridge University Press, S. Bird, B. Boguraev, D. Hindle, M. Kay, D. McDonald, and H. Uszkoreit (2003). *Logics of Conversation*. Studies in Natural Language Processing. Cambridge University Press. ISBN: 9780521650588. URL: https://books.google.fr/books?id=VD-8yisFhBwC (cit. on p. 99).

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). « Latent dirichlet allocation. » In: *Journal of machine Learning research* 3.Jan, pp. 993–1022 (cit. on p. 50).

Erol, B., D.-S. Lee, and J. Hull (2003). « Multimodal Summarization of Meeting Recordings. » In: *Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 3 (ICME '03) - Volume 03*. ICME '03. USA: IEEE Computer Society, pp. 25–28. ISBN: 0780379659 (cit. on p. 99).

Galley, Michel, Kathleen R. McKeown, Eric Fosler-Lussier, and Hongyan Jing (July 2003). « Discourse Segmentation of Multi-Party Conversation. » In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, pp. 562–569. DOI: 10.3115/1075096.1075167. URL: https://www.aclweb.org/anthology/P03-1071 (cit. on pp. 70, 85).

Janin, A. et al. (2003). « The ICSI Meeting Corpus. » In: *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*. Vol. 1, I–364–I–367 vol.1. DOI: 10.1109/ICASSP.2003.1198793 (cit. on pp. 4, 19, 35).

Lin, Chin-Yew and Eduard Hovy (2003). « Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. » In: *Proceedings of the 2003 Human Language Technology Conference of the North Ameri-*

can Chapter of the Association for Computational Linguistics. URL: http://aclweb.org/anthology/N03-1020 (cit. on p. 38).

Grau, Sergio, Emilio Sanchis, Maria Jose Castro, and David Vilar (2004). « Dialogue act classification using a Bayesian approach. » In: *9th Conference Speech and Computer* (cit. on p. 50).

Lin, Chin-Yew (2004). « ROUGE: A Package for Automatic Evaluation of Summaries. » In: *Text Summarization Branches Out*. URL: http://aclweb.org/anthology/W04-1013 (cit. on pp. 5, 17, 38).

Mihalcea, Rada and Paul Tarau (2004). « TextRank: Bringing Order into Text. » In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. URL: http://aclweb.org/anthology/W04-3252 (cit. on pp. 13, 25, 26, 36).

Ang, Jeremy, Yang Liu, and Elizabeth Shriberg (2005). « Automatic dialog act segmentation and classification in multiparty meetings. » In: *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* Vol. 1. IEEE, pp. I–1061 (cit. on p. 50).

Chopra, S., 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)*. Vol. 1, 539–546 vol. 1. DOI: 10.1109/CVPR.2005.202 (cit. on pp. 70, 72, 74, 87).

Evert, Stefan (2005). « The statistics of word cooccurrences. » PhD thesis. Dissertation, Stuttgart University (cit. on p. 13).

LeCun, Yann 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*. Ed. by Robert G. Cowell and Zoubin Ghahramani. Society for Artificial Intelligence and Statistics. URL: http://www.gatsby.ucl.ac.uk/aistats/fullpapers/207.pdf (cit. on pp. 9, 71).

McCowan, Iain, 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*. Vol. 88. URL: http://www.cs.ru.nl/~kraaijw/pubs/Biblio/papers/mccowan-ami-mb2005.pdf (cit. on pp. 3, 4, 19, 35, 68, 83).

Murray, Gabriel, 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*. ISCA, pp. 593–596. URL: http://www.isca-speech.org/archive/interspeech_2005/i05_0593.html (cit. on p. 70).

Venkataraman, Anand, Yang Liu, Elizabeth Shriberg, and Andreas Stolcke (2005). « Does active learning help automatic dialog act tagging

in meeting data? » In: *Ninth European Conference on Speech Communication and Technology* (cit. on pp. 50, 51).

Webb, Nick, Mark Hepple, and Yorick Wilks (2005). « Dialogue act classification based on intra-utterance features. » In: *Proceedings of the AAAI Workshop on Spoken Language Understanding*. Vol. 4. Citeseer, p. 5 (cit. on p. 54).

Csardi, Gabor, Tamas Nepusz, et al. (2006). « The igraph software package for complex network research. » In: *InterJournal, complex systems* 1695.5, pp. 1–9 (cit. on p. 9).

Lecun, Yann, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang (2006). « A tutorial on energy-based learning. » English (US). In: *Predicting structured data*. MIT Press. URL: http://yann.lecun.com/exdb/publis/orig/lecun-06.pdf (cit. on pp. 9, 71, 72).

Liu, Yang (2006). « Using SVM and error-correcting codes for multiclass dialog act classification in meeting corpus. » In: *Ninth International Conference on Spoken Language Processing* (cit. on p. 50).

Morris, Jeremy and Eric Fosler-Lussier (2006). « Combining phonetic attributes using conditional random fields. » In: *Ninth International Conference on Spoken Language Processing* (cit. on p. 52).

Oliphant, Travis E (2006). *A guide to NumPy*. Vol. 1. Trelgol Publishing USA (cit. on p. 9).

Surendran, Dinoj and Gina-Anne Levow (2006). « Dialog act tagging with support vector machines and hidden Markov models. » In: *Ninth International Conference on Spoken Language Processing* (cit. on p. 50).

Gregory, Steve (2007). « An Algorithm to Find Overlapping Community Structure in Networks. » In: *Knowledge Discovery in Databases: PKDD 2007*. Ed. by Joost N. Kok, Jacek Koronacki, Ramon Lopez de Mantaras, Stan Matwin, Dunja Mladenič, and Andrzej Skowron. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 91–102. ISBN: 978-3-540-74976-9 (cit. on p. 86).

Hunter, J. D. (2007). « Matplotlib: A 2D graphics environment. » In: *Computing in Science & Engineering* 9.3, pp. 90–95. DOI: 10.1109/MCSE.2007.55 (cit. on p. 9).

Lendvai, Piroska and Jeroen Geertzen (2007). « Token-based chunking of turn-internal dialogue act sequences. » In: *Proceedings of the 8th SIGDIAL Workshop on Discourse and Dialogue*, pp. 174–181 (cit. on p. 50).

Eisenstein, Jacob and Regina Barzilay (Oct. 2008). « Bayesian Unsupervised Topic Segmentation. » In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 334–343. URL: https://www.aclweb.org/anthology/D08-1035 (cit. on p. 70).

Fernández, Raquel, Matthew Frampton, John Dowding, Anish Adukuzhiyil, Patrick Ehlen, and Stanley Peters (2008). « Identifying relevant phrases to summarize decisions in spoken meetings. » In: *Ninth Annual Con-*

ference of the International Speech Communication Association (cit. on p. 4).

Hagberg, Aric A., Daniel A. Schult, and Pieter J. Swart (2008). « Exploring Network Structure, Dynamics, and Function using NetworkX. » In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, pp. 11 –15 (cit. on p. 9).

Maaten, Laurens van der and Geoffrey Hinton (2008). « Visualizing data using t-SNE. » In: *Journal of machine learning research* 9.Nov, pp. 2579–2605 (cit. on p. 35).

Manning, Christopher D, Hinrich Schütze, and Prabhakar Raghavan (2008). *Introduction to information retrieval*. Cambridge university press (cit. on p. 12).

Murray, Gabriel (2008). « Using speech-specific characteristics for automatic speech summarization. » PhD thesis. Citeseer (cit. on p. 3).

Riedhammer, Korbinian, Dan Gillick, Benoit Favre, and Dilek Hakkani-Tür (2008). « Packing the meeting summarization knapsack. » In: *Ninth Annual Conference of the International Speech Communication Association* (cit. on p. 36).

Tur, Gokhan, Andreas Stolcke, Lynn Voss, John Dowding, Benoît Favre, Raquel Fernández, Matthew Frampton, Michael Frandsen, Clint Frederickson, Martin Graciarena, et al. (2008). « The CALO meeting speech recognition and understanding system. » In: *2008 IEEE Spoken Language Technology Workshop*. IEEE, pp. 69–72 (cit. on p. 2).

Bui, Trung, Matthew Frampton, John Dowding, and Stanley Peters (Sept. 2009). « Extracting Decisions from Multi-Party Dialogue Using Directed Graphical Models and Semantic Similarity. » In: *Proceedings of the SIGDIAL 2009 Conference*. London, UK: Association for Computational Linguistics, pp. 235–243. URL: https://www.aclweb.org/anthology/W09-3934 (cit. on p. 4).

Garg, Nikhil, 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*. ISCA, pp. 1499–1502. URL: http://www.isca-speech.org/archive/interspeech_2009/i09_1499.html (cit. on pp. 36, 70).

Koller, Daphne and Nir Friedman (2009). *Probabilistic graphical models: principles and techniques*. MIT press (cit. on p. 48).

Petukhova, Volha and Harry Bunt (2009). « Who's next? Speaker-selection mechanisms in multiparty dialogue. » In: *Workshop on the Semantics and Pragmatics of Dialogue* (cit. on p. 49).

Zimmermann, Matthias (2009). « Joint segmentation and classification of dialog acts using conditional random fields. » In: *Tenth Annual Conference of the International Speech Communication Association* (cit. on p. 50).

Bunt, Harry et al. (May 2010). « Towards an ISO Standard for Dialogue Act Annotation. » In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/560_Paper.pdf (cit. on p. 47).

Easley, David, Jon Kleinberg, et al. (2010). *Networks, crowds, and markets*. Vol. 8. Cambridge university press Cambridge (cit. on p. 14).

Filippova, Katja (2010). « Multi-Sentence Compression: Finding Shortest Paths in Word Graphs. » In: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, pp. 322–330. URL: http://aclweb.org/anthology/C10-1037 (cit. on pp. 7, 24, 25, 31, 33, 37, 38).

Ganesan, Kavita, ChengXiang Zhai, and Jiawei Han (2010). « Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions. » In: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, pp. 340–348. URL: http://aclweb.org/anthology/C10-1039 (cit. on p. 38).

Kim, Su Nam, Lawrence Cavedon, and Timothy Baldwin (Oct. 2010). « Classifying Dialogue Acts in One-on-One Live Chats. » In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA: Association for Computational Linguistics, pp. 862–871. URL: https://www.aclweb.org/anthology/D10-1084 (cit. on p. 50).

Kitsak, Maksim, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse (2010). « Identification of influential spreaders in complex networks. » In: *Nature Physics* 6.11, pp. 888–893. DOI: 10.1038/nphys1746 (cit. on p. 26).

Lin, Hui and Jeff Bilmes (2010). « Multi-document Summarization via Budgeted Maximization of Submodular Functions. » In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pp. 912–920. URL: http://aclweb.org/anthology/N10-1134 (cit. on pp. 28, 29).

Murray, Gabriel, Giuseppe Carenini, and Raymond Ng (2010). « Generating and Validating Abstracts of Meeting Conversations: a User Study. » In: *Proceedings of the 6th International Natural Language Generation Conference*. URL: https://www.aclweb.org/anthology/W10-4211 (cit. on p. 3).

Newman, Mark (2010). *Networks: An Introduction*. USA: Oxford University Press, Inc. ISBN: 0199206651 (cit. on p. 14).

Řehůřek, Radim and Petr Sojka (May 2010a). « Software Framework for Topic Modelling with Large Corpora. » English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http:

//is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, pp. 45–50 (cit. on p. 9).

Řehůřek, Radim and Petr Sojka (May 2010b). « Software Framework for Topic Modelling with Large Corpora. » English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, pp. 45–50. URL: http://is.muni.cz/publication/884893/en (cit. on p. 56).

Riedhammer, Korbinian, Benoit Favre, and Dilek Hakkani-Tür (Oct. 2010). « Long Story Short - Global Unsupervised Models for Keyphrase Based Meeting Summarization. » In: *Speech Commun.* 52.10, pp. 801–815. ISSN: 0167-6393. DOI: 10.1016/j.specom.2010.06.002. URL: http://dx.doi.org/10.1016/j.specom.2010.06.002 (cit. on p. 30).

Schwämmle, Veit and Ole Nørregaard Jensen (Nov. 2010). « A Simple and Fast Method to Determine the Parameters for Fuzzy C–Means Cluster Analysis. » In: *Bioinformatics* 26.22, pp. 2841–2848. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btq534. URL: http://dx.doi.org/10.1093/bioinformatics/btq534 (cit. on p. 83).

Tur, Gokhan, Andreas Stolcke, Lynn Voss, Stanley Peters, Dilek Hakkani-Tur, John Dowding, Benoit Favre, Raquel Fernández, Matthew Frampton, Mike Frandsen, et al. (2010). « The CALO meeting assistant system. » In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.6, pp. 1601–1611 (cit. on p. 2).

Turney, Peter D and Patrick Pantel (2010). « From frequency to meaning: Vector space models of semantics. » In: *Journal of artificial intelligence research* 37, pp. 141–188 (cit. on p. 13).

Xie, Shasha and Yang Liu (June 2010). « Using Confusion Networks for Speech Summarization. » In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pp. 46–54. URL: https://www.aclweb.org/anthology/N10-1006 (cit. on p. 5).

Carenini, Giuseppe, Gabriel Murray, and Raymond Ng (2011). « Methods for mining and summarizing text conversations. » In: *Synthesis Lectures on Data Management* 3.3, pp. 1–130 (cit. on pp. 3, 11).

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa (2011). « Natural language processing (almost) from scratch. » In: *Journal of machine learning research* 12.Aug, pp. 2493–2537 (cit. on p. 49).

Pedregosa, F. et al. (2011). « Scikit-learn: Machine Learning in Python. » In: *Journal of Machine Learning Research* 12, pp. 2825–2830 (cit. on p. 9).

Tur, Gokhan and Renato De Mori (2011). *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons (cit. on pp. 1, 11).

Van Der Walt, Stefan, S Chris Colbert, and Gael Varoquaux (2011). « The NumPy array: a structure for efficient numerical computation. » In: *Computing in Science & Engineering* 13.2, p. 22 (cit. on p. 9).

Wang, Lu and Claire Cardie (June 2011). « Summarizing Decisions in Spoken Meetings. » In: *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*. Portland, Oregon: Association for Computational Linguistics, pp. 16–24. URL: https://www.aclweb.org/anthology/W11-0503 (cit. on p. 4).

Bunt, Harry, Jan Alexandersson, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David Traum (May 2012). « ISO 24617-2: A semantically-based standard for dialogue annotation. » In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), pp. 430–437. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/530_Paper.pdf (cit. on p. 47).

Lin, Hui (2012). *Submodularity in natural language processing: algorithms and applications*. University of Washington (cit. on pp. 28, 37).

Murray, Gabriel, Giuseppe Carenini, and Raymond Ng (June 2012). « Using the Omega Index for Evaluating Abstractive Community Detection. » In: *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*. Montréal, Canada: Association for Computational Linguistics, pp. 10–18. URL: https://www.aclweb.org/anthology/W12-2602 (cit. on pp. 9, 24, 68, 70, 86, 88).

Sutton, Charles, Andrew McCallum, et al. (2012). « An introduction to conditional random fields. » In: *Foundations and Trends® in Machine Learning* 4.4, pp. 267–373 (cit. on p. 48).

Wang, Lu and Claire Cardie (July 2012). « Focused Meeting Summarization via Unsupervised Relation Extraction. » In: *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Seoul, South Korea: Association for Computational Linguistics, pp. 304–313. URL: https://www.aclweb.org/anthology/W12-1642 (cit. on pp. 4, 7).

Boudin, Florian and Emmanuel Morin (2013). « Keyphrase Extraction for N-best Reranking in Multi-Sentence Compression. » In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 298–305. URL: http://aclweb.org/anthology/N13-1030 (cit. on pp. 25, 33, 34, 37, 38).

Kalchbrenner, Nal and Phil Blunsom (Aug. 2013). « Recurrent Convolutional Neural Networks for Discourse Compositionality. » In: *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 119–126. URL: https://www.aclweb.org/anthology/W13-3214 (cit. on pp. 50, 51).

Mehdad, Yashar, 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*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 136–146. URL: http://aclweb.org/anthology/W13-2117 (cit. on pp. 25, 27, 28, 37, 68, 70, 86).

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013a). « Efficient estimation of word representations in vector space. » In: *arXiv preprint arXiv:1301.3781* (cit. on pp. 14, 56).

Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever (2013). « Exploiting Similarities among Languages for Machine Translation. » In: *CoRR* abs/1309.4168. arXiv: 1309.4168. URL: http://arxiv.org/abs/1309.4168 (cit. on p. 86).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013b). « Distributed representations of words and phrases and their compositionality. » In: *Advances in neural information processing systems*, pp. 3111–3119 (cit. on p. 14).

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig (June 2013). « Linguistic Regularities in Continuous Space Word Representations. » In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 746–751. URL: https://www.aclweb.org/anthology/N13-1090 (cit. on p. 15).

Rousseau, François and Michalis Vazirgiannis (2013). « Graph-of-word and TW-IDF: New Approach to Ad Hoc IR. » In: *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*. CIKM '13. San Francisco, California, USA: ACM, pp. 59–68. ISBN: 978-1-4503-2263-8. DOI: 10.1145/2505515.2505671. URL: http://doi.acm.org/10.1145/2505515.2505671 (cit. on pp. 13, 14, 30).

Tavafi, Maryam, Yashar Mehdad, Shafiq Joty, Giuseppe Carenini, and Raymond Ng (Aug. 2013). « Dialogue Act Recognition in Synchronous and Asynchronous Conversations. » In: *Proceedings of the SIGDIAL 2013 Conference*. Metz, France: Association for Computational Linguistics, pp. 117–121. URL: https://www.aclweb.org/anthology/W13-4017 (cit. on p. 50).

Wang, Lu and Claire Cardie (Aug. 2013). « Domain-Independent Abstract Generation for Focused Meeting Summarization. » In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1395–1405. URL: https://www.aclweb.org/anthology/P13-1137 (cit. on p. 4).

Bae, Joonhyun and Sangwook Kim (2014). « Identifying and ranking influential spreaders in complex networks by neighborhood coreness. » In: *Physica A: Statistical Mechanics and its Applications* 395, pp. 549–559 (cit. on p. 27).

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). « Neural machine translation by jointly learning to align and translate. » In: *arXiv preprint arXiv:1409.0473* (cit. on p. 5).

Baroni, Marco, Georgiana Dinu, and Germán Kruszewski (June 2014). « Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. » In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 238–247. DOI: 10.3115/v1/P14-1023. URL: https://www.aclweb.org/anthology/P14-1023 (cit. on p. 13).

Cho, Kyunghyun, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (Oct. 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)*. Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. URL: https://www.aclweb.org/anthology/D14-1179 (cit. on pp. 16, 79).

Higashinaka, Ryuichiro, Kenji Imamura, Toyomi Meguro, Chiaki Miyazaki, Nozomi Kobayashi, Hiroaki Sugiyama, Toru Hirano, Toshiro Makino, and Yoshihiro Matsuo (Aug. 2014). « Towards an open-domain conversational system fully based on natural language processing. » In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, pp. 928–939. URL: https://www.aclweb.org/anthology/C14-1088 (cit. on p. 48).

Jurafsky, D., J.H. Martin, P. Norvig, and S. Russell (2014). *Speech and Language Processing*. Pearson Education. ISBN: 9780133252934. URL: https://books.google.fr/books?id=Cq2gBwAAQBAJ (cit. on p. 11).

Merkel, Dirk (2014). « Docker: lightweight linux containers for consistent development and deployment. » In: *Linux journal* 2014.239, p. 2 (cit. on p. 2).

Milajevs, Dmitrijs and Matthew Purver (Apr. 2014). « Investigating the Contribution of Distributional Semantic Information for Dialogue Act Classification. » In: *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 40–47. DOI: 10.3115/v1/W14-1505. URL: https://www.aclweb.org/anthology/W14-1505 (cit. on p. 54).

Oya, Tatsuro, 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)*. Philadelphia, Pennsylvania, U.S.A.: Association for Computational

Linguistics, pp. 45–53. DOI: 10.3115/v1/W14-4407. URL: http://aclweb.org/anthology/W14-4407 (cit. on pp. 4, 68, 70, 85).

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). « Dropout: A Simple Way to Prevent Neural Networks from Overfitting. » In: *Journal of Machine Learning Research* 15, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html (cit. on p. 87).

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). « Sequence to sequence learning with neural networks. » In: *Advances in neural information processing systems*, pp. 3104–3112 (cit. on p. 5).

Wang, Jiang, 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. Washington, DC, USA: IEEE Computer Society, pp. 1386–1393. ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.180. URL: https://doi.org/10.1109/CVPR.2014.180 (cit. on pp. 73, 87).

Wang, Rui, Wei Liu, and Chris McDonald (2014). « Corpus-independent generic keyphrase extraction using word embedding vectors. » In: *Software Engineering Research Conference*. Vol. 39 (cit. on pp. 28, 33).

Abadi, Martín et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: https://www.tensorflow.org/ (cit. on p. 9).

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). « Neural Machine Translation by Jointly Learning to Align and Translate. » In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1409.0473 (cit. on p. 80).

Banerjee, Siddhartha, Prasenjit Mitra, and Kazunari Sugiyama (2015). « Generating Abstractive Summaries from Meeting Transcripts. » In: *Proceedings of the 2015 ACM Symposium on Document Engineering*. DocEng '15. Lausanne, Switzerland: ACM, pp. 51–60. ISBN: 978-1-4503-3307-8. DOI: 10.1145/2682571.2797061. URL: http://doi.acm.org/10.1145/2682571.2797061 (cit. on pp. 68, 70, 85).

Chollet, François et al. (2015). *Keras*. https://keras.io (cit. on p. 9).

Ganesan, Kavita (2015). « ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks. » In: (cit. on p. 10).

Hoffer, Elad 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*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6622 (cit. on pp. 9, 70, 73, 87).

Huang, Zhiheng, Wei Xu, and Kai Yu (2015). « Bidirectional LSTM-CRF models for sequence tagging. » In: *arXiv preprint arXiv:1508.01991* (cit. on pp. 48–50).

Kingma, Diederik P. 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*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6980 (cit. on pp. 56, 87).

Kusner, Matt J., Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger (2015). « From Word Embeddings to Document Distances. » In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, pp. 957–966 (cit. on p. 29).

Luong, Thang, Hieu Pham, and Christopher D. Manning (Sept. 2015). « Effective Approaches to Attention-based Neural Machine Translation. » In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1412–1421. DOI: 10.18653/v1/D15-1166. URL: https://www.aclweb.org/anthology/D15-1166 (cit. on p. 80).

Meladianos, Polykarpos, Giannis Nikolentzos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis (2015). « Degeneracy-based real-time sub-event detection in twitter stream. » In: *ICWSM* 15, pp. 248–257 (cit. on p. 14).

Ng, Jun-Ping and Viktoria Abrecht (Sept. 2015). « Better Summarization Evaluation with Word Embeddings for ROUGE. » In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1925–1930. DOI: 10.18653/v1/D15-1222. URL: https://www.aclweb.org/anthology/D15-1222 (cit. on p. 5).

Rousseau, François (2015). « Graph-of-words: mining and retrieving text with networks of features. » PhD thesis. Ph. D. Dissertation. École Polytechnique (cit. on p. 14).

Rousseau, François and Michalis Vazirgiannis (2015). « Main core retention on graph-of-words for single-document keyword extraction. » In: *European Conference on Information Retrieval*. Springer, pp. 382–393 (cit. on p. 14).

Rush, Alexander M., Sumit Chopra, and Jason Weston (Sept. 2015). « A Neural Attention Model for Abstractive Sentence Summarization. » In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 379–389. DOI: 10.18653/v1/D15-1044. URL: https://www.aclweb.org/anthology/D15-1044 (cit. on p. 5).

Schroff, F., 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)*. Vol. 00, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682. URL: doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298682 (cit. on pp. 70, 73, 87).

Amos, Brandon, Bartosz Ludwiczuk, and Mahadev Satyanarayanan (2016). *OpenFace: A general-purpose face recognition library with mobile*

applications. Tech. rep. CMU-CS-16-118, CMU School of Computer Science. URL: http://elijah.cs.cmu.edu/DOCS/CMU-CS-16-118.pdf (cit. on p. 87).

Bishop, C.M. (2016). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer New York. ISBN: 9781493938438. URL: https://books.google.fr/books?id=kOXDtAEACAAJ (cit. on p. 11).

Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio (2016). *Deep learning*. Vol. 1. MIT press Cambridge (cit. on pp. 5, 11).

Khanpour, Hamed, Nishitha Guntakandla, and Rodney Nielsen (Dec. 2016). « Dialogue Act Classification in Domain-Independent Conversations Using a Deep Recurrent Neural Network. » In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 2012–2021. URL: https://www.aclweb.org/anthology/C16-1189 (cit. on p. 50).

Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer (June 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*. San Diego, California: Association for Computational Linguistics, pp. 260–270. DOI: 10.18653/v1/N16-1030. URL: https://www.aclweb.org/anthology/N16-1030 (cit. on pp. 48, 49).

Lee, Ji Young 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*. San Diego, California: Association for Computational Linguistics, pp. 515–520. DOI: 10.18653/v1/N16-1062. URL: http://aclweb.org/anthology/N16-1062 (cit. on pp. 50, 54, 84).

Li, Wei and Yunfang Wu (Dec. 2016). « Multi-level Gated Recurrent Neural Network for dialog act classification. » In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 1970–1979. URL: https://www.aclweb.org/anthology/C16-1185 (cit. on pp. 48, 50, 51).

Mueller, Jonas 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*. Ed. by Dale Schuurmans and Michael P. Wellman. AAAI Press, pp. 2786–2792. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12195 (cit. on pp. 9, 70, 72, 73).

Nallapati, Ramesh, Bowen Zhou, Cicero dos Santos, Çağlar GuÌ‡lçehre, and Bing Xiang (Aug. 2016). « Abstractive Text Summarization us-

ing Sequence-to-sequence RNNs and Beyond. » In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, pp. 280–290. DOI: 10.18653/v1/K16-1028. URL: https://www.aclweb.org/anthology/K16-1028 (cit. on p. 5).

Neculoiu, Paul, Maarten Versteegh, and Mihai Rotaru (2016). « Learning Text Similarity with Siamese Recurrent Networks. » In: *Proceedings of the 1st Workshop on Representation Learning for NLP*. Berlin, Germany: Association for Computational Linguistics, pp. 148–157. DOI: 10.18653/v1/W16-1617. URL: http://aclweb.org/anthology/W16-1617 (cit. on pp. 74, 87).

Shen, Sheng-syun and Hung-yi Lee (2016). « Neural attention models for sequence classification: Analysis and application to key term extraction and dialogue act detection. » In: *arXiv preprint arXiv:1604.00077* (cit. on p. 50).

Tixier, Antoine, Fragkiskos Malliaros, and Michalis Vazirgiannis (2016). « A Graph Degeneracy-based Approach to Keyword Extraction. » In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1860–1870. DOI: 10.18653/v1/D16-1191. URL: http://aclweb.org/anthology/D16-1191 (cit. on pp. 14, 25–27).

Tixier, Antoine, Konstantinos Skianis, and Michalis Vazirgiannis (Aug. 2016). « GoWvis: A Web Application for Graph-of-Words-based Text Visualization and Summarization. » In: *Proceedings of ACL-2016 System Demonstrations*. Berlin, Germany: Association for Computational Linguistics, pp. 151–156. DOI: 10.18653/v1/P16-4026. URL: https://www.aclweb.org/anthology/P16-4026 (cit. on pp. 13, 26).

Tu, Zhaopeng, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li (2016). « Modeling coverage for neural machine translation. » In: *arXiv preprint arXiv:1601.04811* (cit. on p. 81).

Yang, Zichao, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy (June 2016a). « 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*. San Diego, California: Association for Computational Linguistics, pp. 1480–1489. DOI: 10.18653/v1/N16-1174. URL: https://www.aclweb.org/anthology/N16-1174 (cit. on p. 65).

Yang, Zichao, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy (2016b). « 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*. San Diego, California: Association for Computational Linguistics, pp. 1480–1489. DOI: 10.18653/v1/N16-1174. URL: http://aclweb.org/anthology/N16-1174 (cit. on pp. 80, 84).

Gambhir, Mahak and Vishal Gupta (2017). « Recent automatic text summarization techniques: a survey. » In: *Artificial Intelligence Review* 47.1, pp. 1–66 (cit. on p. 2).

Kim, Geonmin, Hwaran Lee, Bokyeong Kim, and Soo-young Lee (2017). « Compositional Sentence Representation from Character Within Large Context Text. » In: *Neural Information Processing*. Ed. by Derong Liu, Shengli Xie, Yuanqing Li, Dongbin Zhao, and El-Sayed M. El-Alfy. Cham: Springer International Publishing, pp. 674–685. ISBN: 978-3-319-70096-0 (cit. on p. 54).

Lin, Zhouhan, 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*. URL: https://openreview.net/forum?id=BJC_jUqxe (cit. on p. 80).

Liu, Yang, Kun Han, Zhao Tan, and Yun Lei (Sept. 2017). « Using Context Information for Dialog Act Classification in DNN Framework. » In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2170–2178. DOI: 10.18653/v1/D17-1231. URL: https://www.aclweb.org/anthology/D17-1231 (cit. on pp. 48, 50, 51, 56, 60).

Meladianos, Polykarpos, Antoine Tixier, Ioannis Nikolentzos, and Michalis Vazirgiannis (2017). « Real-Time Keyword Extraction from Conversations. » In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 462–467. URL: http://aclweb.org/anthology/E17-2074 (cit. on pp. 14, 27, 48).

Ortega, Daniel and Ngoc Thang Vu (Aug. 2017). « Neural-based Context Representation Learning for Dialog Act Classification. » In: *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. Saarbrücken, Germany: Association for Computational Linguistics, pp. 247–252. DOI: 10.18653/v1/W17-5530. URL: https://www.aclweb.org/anthology/W17-5530 (cit. on p. 50).

Reimers, Nils and Iryna Gurevych (2017). « Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. » In: *arXiv preprint arXiv:1707.06799* (cit. on pp. 50, 60).

See, Abigail, Peter J. Liu, and Christopher D. Manning (July 2017a). « 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)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1073–1083. DOI: 10.18653/v1/P17-1099. URL: https://www.aclweb.org/anthology/P17-1099 (cit. on p. 5).

See, Abigail, Peter J. Liu, and Christopher D. Manning (2017b). « 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)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1073–1083. DOI: 10.18653/v1/P17-1099. URL: http://aclweb.org/anthology/P17-1099 (cit. on p. 81).

Singla, Karan, 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*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 43–47. DOI: 10.18653/v1/W17-4506. URL: http://aclweb.org/anthology/W17-4506 (cit. on pp. 70, 85).

Tixier, Antoine, 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*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 48–58. DOI: 10.18653/v1/W17-4507. URL: http://aclweb.org/anthology/W17-4507 (cit. on pp. 14, 36, 70).

Tran, Quan Hung, Ingrid Zukerman, and Gholamreza Haffari (Apr. 2017). « A Hierarchical Neural Model for Learning Sequences of Dialogue Acts. » In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 428–437. URL: https://www.aclweb.org/anthology/E17-1041 (cit. on pp. 48, 50).

Vaswani, Ashish, 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*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, pp. 6000–6010. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need (cit. on pp. 5, 80).

Wang, Longyue, Zhaopeng Tu, Andy Way, and Qun Liu (2017). « Exploiting cross-sentence context for neural machine translation. » In: *arXiv preprint arXiv:1704.04347*. URL: https://www.aclweb.org/anthology/D17-1301 (cit. on p. 79).

Alizadeh, Pegah, Peggy Cellier, Thierry Charnois, Bruno Crémilleux, and Albrecht Zimmermann (Mar. 2018). « An Experimental Approach For Information Extraction in Multi-Party Dialogue Discourse. » In: *CICLing 2018 - 19th International Conference on Computational Linguis-*

tics and Intelligent Text Processing. Hanoi, Vietnam, pp. 1–14. URL: https://hal.archives-ouvertes.fr/hal-01804147 (cit. on p. 2).

Bothe, Chandrakant, Cornelius Weber, Sven Magg, and Stefan Wermter (May 2018). « A Context-based Approach for Dialogue Act Recognition using Simple Recurrent Neural Networks. » In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://www.aclweb.org/anthology/L18-1307 (cit. on pp. 50, 51, 56, 60).

Chen, Zheqian, Rongqin Yang, Zhou Zhao, Deng Cai, and Xiaofei He (2018). « Dialogue Act Recognition via CRF-Attentive Structured Network. » In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR '18. Ann Arbor, MI, USA: Association for Computing Machinery, pp. 225–234. ISBN: 9781450356572. DOI: 10.1145/3209978.3209997. URL: https://doi.org/10.1145/3209978.3209997 (cit. on pp. 48–50, 57).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). « Bert: Pre-training of deep bidirectional transformers for language understanding. » In: *arXiv preprint arXiv:1810.04805* (cit. on p. 100).

Kumar, Harshit, Arvind Agarwal, Riddhiman Dasgupta, and Sachindra Joshi (2018). « Dialogue Act Sequence Labeling Using Hierarchical Encoder With CRF. » In: *AAAI Conference on Artificial Intelligence*. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16706 (cit. on pp. 48–50, 56, 57, 65).

Luo, Ling, Zhihao Yang, Pei Yang, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang (2018). « An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. » In: *Bioinformatics* 34.8, pp. 1381–1388 (cit. on p. 65).

Pan, Haojie, Junpei Zhou, Zhou Zhao, Yan Liu, Deng Cai, and Min Yang (2018). « Dial2Desc: End-to-end Dialogue Description Generation. » In: *CoRR* abs/1811.00185. arXiv: 1811.00185. URL: http://arxiv.org/abs/1811.00185 (cit. on p. 96).

Shang, Guokan, 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)*. Melbourne, Australia: Association for Computational Linguistics, pp. 664–674. URL: http://aclweb.org/anthology/P18-1062 (cit. on pp. 48, 68, 70).

Su, Shang-Yu, Pei-Chieh Yuan, and Yun-Nung Chen (June 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). New Orleans, Louisiana: Association for Computational Linguistics, pp. 2133–2142. DOI: 10.18653/v1/N18-1194. URL: https://www.aclweb.org/anthology/N18-1194 (cit. on p. 82).

Yang, Jie, Shuailong Liang, and Yue Zhang (2018). « Design challenges and misconceptions in neural sequence labeling. » In: *arXiv preprint arXiv:1806.04470* (cit. on pp. 50, 60).

Zhou, Jie, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun (2018). « Graph neural networks: A review of methods and applications. » In: *arXiv preprint arXiv:1812.08434* (cit. on p. 99).

Ahmadvand, Ali, Jason Ingyu Choi, and Eugene Agichtein (2019). « Contextual Dialogue Act Classification for Open-Domain Conversational Agents. » In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1273–1276 (cit. on p. 48).

Chen, Lingzhen and Alessandro Moschitti (2019). « Transfer learning for sequence labeling using source model and target data. » In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 6260–6267 (cit. on p. 52).

Chen, Qian, Zhu Zhuo, and Wen Wang (2019). « Bert for joint intent classification and slot filling. » In: *arXiv preprint arXiv:1902.10909* (cit. on p. 65).

Cui, Leyang and Yue Zhang (2019). « Hierarchically-Refined Label Attention Network for Sequence Labeling. » In: *arXiv preprint arXiv:1908.08676* (cit. on pp. 50, 60).

Gliwa, Bogdan, Iwona Mochol, Maciej Biesek, and Aleksander Wawer (Nov. 2019). « SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. » In: *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Hong Kong, China: Association for Computational Linguistics, pp. 70–79. DOI: 10.18653/v1/D19-5409. URL: https://www.aclweb.org/anthology/D19-5409 (cit. on p. 96).

Jacquenet, François, Marc Bernard, and Christine Largeron (2019). « Meeting summarization, A challenge for deep learning. » In: *International Work-Conference on Artificial Neural Networks*. Springer, pp. 644–655 (cit. on p. 2).

Kryscinski, Wojciech, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher (Nov. 2019). « Neural Text Summarization: A Critical Evaluation. » In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 540–551. DOI: 10.18653/v1/D19-1051. URL: https://www.aclweb.org/anthology/D19-1051 (cit. on p. 5).

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer

(2019). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. arXiv: 1910.13461 [cs.CL] (cit. on p. 5).

Li, Manling, Lingyu Zhang, Heng Ji, and Richard J. Radke (July 2019a). « Keep Meeting Summaries on Topic: Abstractive Multi-Modal Meeting Summarization. » In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 2190–2196. DOI: 10.18653/v1/P19-1210. URL: https://www.aclweb.org/anthology/P19-1210 (cit. on pp. 5, 99).

Li, Ruizhe, Chenghua Lin, Matthew Collinson, Xiao Li, and Guanyi Chen (Nov. 2019b). « A Dual-Attention Hierarchical Recurrent Neural Network for Dialogue Act Classification. » In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, pp. 383–392. DOI: 10.18653/v1/K19-1036. URL: https://www.aclweb.org/anthology/K19-1036 (cit. on pp. 48–50, 65).

Liu, Jinchao, Stuart J. Gibson, James Mills, and Margarita Osadchy (2019). « Dynamic spectrum matching with one-shot learning. » In: *Chemometrics and Intelligent Laboratory Systems* 184, pp. 175 –181. ISSN: 0169-7439. DOI: 10.1016/j.chemolab.2018.12.005. URL: http://www.sciencedirect.com/science/article/pii/S0169743918304805 (cit. on p. 87).

Liu, Yang and Mirella Lapata (Nov. 2019). « Text Summarization with Pretrained Encoders. » In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3730–3740. DOI: 10.18653/v1/D19-1387. URL: https://www.aclweb.org/anthology/D19-1387 (cit. on p. 100).

Lorré, Jean-Pierre, Isabelle Ferrané, Francisco Madrigal, Michalis Vazirgiannis, and Christophe Bourguignat (2019). « LinTO: Assistant vocal open-source respectueux des données personnelles pour les réunions d'entreprise. » In: *APIA*, p. 63 (cit. on p. 2).

Lu, Yang (2019). « Artificial intelligence: a survey on evolution, models, applications and future trends. » In: *Journal of Management Analytics* 6.1, pp. 1–29 (cit. on p. 1).

Nedoluzhko, Anna and Ondrej Bojar (2019). « Towards Automatic Minuting of the Meetings. » In: *ITAT*, pp. 112–119 (cit. on p. 4).

Nikolentzos, Giannis, Antoine J-P Tixier, and Michalis Vazirgiannis (2019). « Message Passing Attention Networks for Document Understanding. » In: *arXiv preprint arXiv:1908.06267* (cit. on p. 14).

Raheja, Vipul and Joel Tetreault (June 2019). « Dialogue Act Classification with Context-Aware Self-Attention. » 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: Association for Computational Linguistics, pp. 3727–3733. DOI: 10.18653/v1/N19-1373. URL: https://www.aclweb.org/anthology/N19-1373 (cit. on pp. 48–50, 54, 65).

Ruder, Sebastian, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf (June 2019). « Transfer Learning in Natural Language Processing. » In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 15–18. DOI: 10.18653/v1/N19-5004. URL: https://www.aclweb.org/anthology/N19-5004 (cit. on p. 100).

Shang, Guokan, Antoine Jean-Pierre Tixier, Michalis Vazirgiannis, and Jean-Pierre Lorré (2019). « Energy-based self-attentive learning of abstractive communities for spoken language understanding. » In: *arXiv preprint arXiv:1904.09491* (cit. on p. 48).

Winata, Genta Indra, Zhaojiang Lin, Jamin Shin, Zihan Liu, and Pascale Fung (2019). « Hierarchical Meta-Embeddings for Code-Switching Named Entity Recognition. » In: *arXiv preprint arXiv:1909.08504* (cit. on p. 65).

Xu, Jiacheng, Zhe Gan, Yu Cheng, and Jingjing Liu (2019). « Discourse-aware neural extractive model for text summarization. » In: *arXiv preprint arXiv:1910.14142* (cit. on p. 99).

Yan, Hang, Bocao Deng, Xiaonan Li, and Xipeng Qiu (2019). « TENER: Adapting Transformer Encoder for Name Entity Recognition. » In: *arXiv preprint arXiv:1911.04474* (cit. on p. 65).

Zhang, Linhao and Houfeng Wang (2019). « Using Bidirectional Transformer-CRF for Spoken Language Understanding. » In: *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, pp. 130–141 (cit. on p. 65).

Chapuis, Emile, Pierre Colombo, Matteo Manica, Matthieu Labeau, and Chloé Clavel (Nov. 2020). « Hierarchical Pre-training for Sequence Labelling in Spoken Dialog. » In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 2636–2648. DOI: 10.18653/v1/2020.findings-emnlp.239. URL: https://www.aclweb.org/anthology/2020.findings-emnlp.239 (cit. on p. 50).

Colombo, Pierre, Emile Chapuis, Matteo Manica, Emmanuel Vignon, Giovanna Varni, and Chloe Clavel (2020). « Guiding Attention in Sequence-to-Sequence Models for Dialogue Act Prediction. » In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05, pp. 7594–7601. DOI: 10.1609/aaai.v34i05.6259. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6259 (cit. on pp. 50, 65).

Doan, Tu My, Francois Jacquenet, Christine Largeron, and Marc Bernard (2020). « A Study of Text Summarization Techniques for Generating Meeting Minutes. » In: *International Conference on Research Challenges in Information Science*. Springer, pp. 522–528 (cit. on p. 2).

Gururangan, Suchin, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith (July 2020). « Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. » In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8342–8360. DOI: 10.18653/v1/2020.acl-main.740. URL: https://www.aclweb.org/anthology/2020.acl-main.740 (cit. on p. 100).

Rebai, Ilyes, Sami Benhamiche, Kate Thompson, Zied Sellami, Damien Laine, and Jean-Pierre Lorré (May 2020). « LinTO Platform: A Smart Open Voice Assistant for Business Environments. » English. In: *Proceedings of the 1st International Workshop on Language Technology Platforms*. Marseille, France: European Language Resources Association, pp. 89–95. ISBN: 979-10-95546-64-1. URL: https://www.aclweb.org/anthology/2020.iwltp-1.14 (cit. on p. 2).

Zhu, Chenguang, Ruochen Xu, Michael Zeng, and Xuedong Huang (2020). *End-to-End Abstractive Summarization for Meetings*. arXiv: 2004.02016 [cs.CL] (cit. on p. 5).

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis`. Most of the graphics in this dissertation are generated using the `Matplotlib` library for the Python programming language. The bibliography is typeset using the `biblatex`.

ECOLE DOCTORALE

Titre : Compréhension du Langage Parlé pour le Résumé Abstractif de Réunion

Mots clés : résumé abstractif de réunion, résumé automatique de texte, compréhension du langage parlé, traitement automatique des langues, apprentissage automatique, intelligence artificielle

Résumé : Grâce aux progrès impressionnants qui ont été réalisés dans la transcription du langage parlé, il est de plus en plus possible d'exploiter les données transcrites pour des tâches qui requièrent la compréhension de ce que l'on dit dans une conversation. Le travail présenté dans cette thèse, réalisé dans le cadre d'un projet consacré au développement d'un assistant de réunion, contribue aux efforts en cours pour apprendre aux machines à comprendre les dialogues des réunions multipartites. Nous nous sommes concentrés sur le défi de générer automatiquement les résumés abstractifs de réunion.

Nous présentons tout d'abord nos résultats sur le Résumé Abstractif de Réunion (RAR), qui consiste à prendre une transcription de réunion comme entrée et à produire un résumé abstractif comme sortie. Nous introduisons une approche entièrement non-supervisée pour cette tâche, basée sur la compression multi-phrases et la maximisation sous-modulaire budgétisée. Nous tirons également parti des progrès récents en vecteurs de mots et dégénérescence de graphes appliqués au TAL, afin de prendre en compte les connaissances sémantiques extérieures et de concevoir de nouvelles mesures de diversité et d'informativité.

Ensuite, nous discutons de notre travail sur la Classification en Actes de Dialogue (CAD), dont le but est d'attribuer à chaque énoncé d'un discours une étiquette qui représente son intention communicative. La CAD produit des annotations qui sont utiles pour une grande variété de tâches, y compris le RAR. Nous proposons une couche neuronale modifiée de Champ Aléatoire Conditionnel (CAC) qui prend en compte non seulement la séquence des énoncés dans un discours, mais aussi les informations sur les locuteurs et en particulier, s'il y a eu un changement de locuteur d'un énoncé à l'autre.

La troisième partie de la thèse porte sur la Détection de Communauté Abstractive (DCA), une sous-tâche du RAR, dans laquelle les énoncés d'une conversation sont regroupés selon qu'ils peuvent être résumés conjointement par une phrase abstractive commune. Nous proposons une nouvelle approche de la DCA dans laquelle nous introduisons d'abord un encodeur neuronal contextuel d'énoncé qui comporte trois types de mécanismes d'auto-attention, puis nous l'entraînons en utilisant les méta-architectures siamoise et triplette basées sur l'énergie. Nous proposons en outre une méthode d'échantillonnage générale qui permet à l'architecture triplette de capturer des motifs subtils (p. ex., des groupes qui se chevauchent et s'emboîtent).

Title : Spoken Language Understanding for Abstractive Meeting Summarization

Keywords : abstractive meeting summarization, automatic text summarization, spoken language understanding, natural language processing, machine learning, artificial intelligence

Abstract : With the impressive progress that has been made in transcribing spoken language, it is becoming increasingly possible to exploit transcribed data for tasks that require comprehension of what is said in a conversation. The work in this dissertation, carried out in the context of a project devoted to the development of a meeting assistant, contributes to ongoing efforts to teach machines to understand multi-party meeting speech. We have focused on the challenge of automatically generating abstractive meeting summaries.

We first present our results on Abstractive Meeting Summarization (AMS), which aims to take a meeting transcription as input and produce an abstractive summary as output. We introduce a fully unsupervised framework for this task based on multi-sentence compression and budgeted submodular maximization. We also leverage recent advances in word embeddings and graph degeneracy applied to NLP, to take exterior semantic knowledge into account and to design custom diversity and informativeness measures.

Next, we discuss our work on Dialogue Act Classification (DAC), whose goal is to assign each utterance in a discourse a label that represents its communicative intention. DAC yields annotations that are useful for a wide variety of tasks, including AMS. We propose a modified neural Conditional Random Field (CRF) layer that takes into account not only the sequence of utterances in a discourse, but also speaker information and in particular, whether there has been a change of speaker from one utterance to the next.

The third part of the dissertation focuses on Abstractive Community Detection (ACD), a sub-task of AMS, in which utterances in a conversation are grouped according to whether they can be jointly summarized by a common abstractive sentence. We provide a novel approach to ACD in which we first introduce a neural contextual utterance encoder featuring three types of self-attention mechanisms and then train it using the siamese and triplet energy-based meta-architectures. We further propose a general sampling scheme that enables the triplet architecture to capture subtle patterns (e.g., overlapping and nested clusters).

Institut Polytechnique de Paris
91120 Palaiseau, France