The Interdisciplinary Center, Herzlia
Efi Arazi School of Computer Science
M.Sc. program - Research Track

Opinionated Natural Language Generation

by
Tomer Cagan

M.Sc. dissertation, submitted in partial fulfillment of the requirements for the M.Sc. degree, research track, School of Computer Science
The Interdisciplinary Center, Herzliya

August 2016

This work was carried out under the supervision of Dr. Reut Tsarfaty from the Dept. of Mathematics and Computer Science, at The Open University, Ra'anana, and Prof. Ariel Shamir from the Efi Arazi School of Computer Science, The Interdisciplinary Center, Herzliya.

"*The wise are not wise because they make no mistakes. They are wise because they correct their mistakes as soon as they recognize them.*"

Orson Scott Card, Xenocide

"*Knowledge is just an opinion that you trust enough to act upon.*"

Orson Scott Card, Children Of The Mind

"*Test can't measure what really matters.*"

Orson Scott Card, Speaker for the Dead

Abstract

Opinionated, Natural Language Generation (NLG)

by Tomer CAGAN

In this work we address natural language generation (NLG) in the context of social media. As more of today's communication is being conducted online, the importance of understanding and communicating with online users is becoming increasingly important for businesses, governments and non-profit organizations. Furthermore, a lot of casual, day-to-day, interpersonal interaction has moved from traditional mediums to the virtual world, which in turn places more significance on the research of online interactions.

In this work we aim to create computer generated texts that seem natural and human-like while also being opinionated and expressive. Relying on state of the art technique for analyzing and understanding texts, concentrating on sentiment and topics, we define and build a user model for an online responder and then explore different ways to generate human-like and relevant responses.

First, we build a template-based system that automatically generates replies to online documents. The system uses hand-crafted grammatical templates with placeholders for referring expressions, which are dynamically filled according to the relevant online context. We present a Turing-test like method of evaluating the resulting responses and show that we could get close to human responses quality. Specifically, we show that including world-knowledge in response generation increases its human-likeness and that responses with a more positive sentiment are considered more computer-like. On the other hand, we empirically observed a clear learning effect – where human readers learn to identify computer-generated responses over time. This effect stems from the low variance of the template-based approach.

To address the shortcoming of the template-based approach, and in particular, its low variance and close tie to specific domains, we designed a data-driven system that is based on a grammar-based generation architecture. We develop several types of grammars for generation; (i) a simple probabilistic context free grammar (PCFG), (ii) a lexicalized grammar akin to Collins (1997), and (iii) a relational-realizational grammar (RR), based on Tsarfaty and Sima'an (2008).

We compared the grammars using a similar Turing-like evaluation test and automatically evaluated compactness, fluency and sentiment agreement of the responses. We find that the relational-realizational grammar is more compact and yields better responses when evaluated with language model. The lexicalized grammar shows higher sentiment agreement but outscores the RR grammar by small margins. Next, we showed that by including the topic model in the response generation, we are able to get more relevant responses. In online human-likeness evaluation survey we get a slightly different results in which the lexicalized grammar out-performs the other two grammars.

The contribution of this thesis is hence manifold. We introduce a novel task of *opinionated NLG*, and provide 2 general architectures for generation of opinionated responses. We release a new decorated dataset for inducing grammars, and introduce novel evaluation methodologies. Our results provide new insights concerning key differences between human-generated and computer-generated responses in the

hope of inspiring further research and more sophisticated modeling of Opinionated NLG research.

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Reut Tsarfaty, who guided me through this process. Challenging with new ideas and directing with a sure hand – I am really happy I had this opportunity – it was interesting and stimulating.

Next, I would like to express my gratitude and acknowledge the contribution of Prof. Stefan Frank who helped us devise the human evaluation and analyze the results – the discussion following each survey was intriguing and his nitpicking made our results so much better.

Thanks are also due to the Efi Arazi School of Computer Science faculty at the Interdisciplinary Center – to Prof. Ariel Shamir who was my internal supervisor and was available whenever we needed his advice; to Prof. Tami Tamir who supported the publication of the first phase of this work; and the rest of the faculty and stuff who made this (long) journey meaningful and enjoyable. Also, to Dr. Doron Friedman from the Sammy Ofer School of Communication who was willing to share his views and knowledge from a different perspective.

The work presented in Chapter 4 has been published in the context of Cagan, Frank, and Tsarfaty (2014) at the Joint Workshop on Social Dynamics and Personal Attributes in Social Media. I want to thank the anonymous reviewers as well as the audience and organizers of this meeting for the insightful feedback and interesting discussion. In addition, we would like to thanks the anonymous reviewers of ACL2014, WASSA2014 and COLING2014, whose feedback also help us improved our work.

And of course, to my family and close friends who may have missed my presence in the long period it took to get this research completed – I sure did missed you all.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Written texts in digital media are quickly becoming the prevalent, if not preferred, communication method of many people nowadays. Social network traffic (posts, tweets), comments in news sites, and the ever present chat applications, are all common examples of interpersonal communication conducted online. These new media, and general online user-generated content, is enabling effective human interaction; so much so that many of our day-to-day interactions are conducted online (Viswanath et al., 2009).

Such online interaction in social media fundamentally changes the way businesses and consumers behave (Qualman, 2012), can be instrumental to the success of individuals and businesses (Haenlein and Kaplan, 2009), and even affects the stability of political regimes (Howard et al., 2011; Lamer, 2012). This state of affairs forces organizations (businesses, governments, and non-profit institutions) to be constantly involved in the monitoring of, and the interaction with, human agents in digital environments (Langheinrich and Karjoth, 2011).

Automatic analysis of user-generated online content benefits from extensive research and commercial opportunities. In natural language processing, there is ample research on the analysis of subjectivity and sentiment of content in social media. The development of tools for sentiment analysis (Davidov, Tsur, and Rappoport, 2010), opinion mining (Mishne, 2006), and many more, currently enjoys a wide interest and exposure[1].

Unlike the analysis efforts, we see less work on *generating* texts in the context of social media and online interaction. A study by Ritter, Cherry, and Dolan (2011) addresses the generation of responses to short natural language texts (tweets) as a machine-translation problem in a data-driven setup, and uses user surveys for evaluation. A study by Hasegawa et al. (2013) modifies Ritter's approach, adding the goal of producing responses that should elicit emotions in the addressees. In both studies, the generated responses echo-out the original text and do not target particular topics. Furthermore, the generated responses do not carry explicit user characteristics and opinions.

Much generation research is often based on some database or knowledge system and produces technical texts (Lester, 1994). Additional research dealing with natural language generation and human–computer interfaces also exhibits a non-personal nature. For example, software for *Augmentative and Alternative Communication* (AAC, Dempster, Alm, and Reiter (2010)) aids handicap people in conducting dialogs, and *Chat–bots* (Mori, Jatowt, and Ishizuka, 2003; Feng et al., 2006) automatically interact with users for specific task, both generally concentrate on more technical aspects of language.

[1] As is also evident by the many workshops and dedicated tracks at ACL venues. E.g., the ACL series LASM http://tinyurl.com/z6mcfy9; WASSA http://optima.jrc.it/wassa2016/.

An additional emerging human-computer interaction field of research, that of *Intelligent Personal Assistant*, such as Apple's Siri or Microsoft's Cortana, is more concerned with development of knowledge representation (Chaudhri et al., 2006) and discourse understanding (Niekrasz et al., 2005) and less with the communication.

Research of natural language analysis contributes to *understanding* online communication and online users. In order to *interact* with the users, similar efforts are required for natural language generation. Meaningful written interaction in social media should relate to content from the media itself and should consist of personal and opinionated texts. As with any communication effort, there is a *communicative goal* which the author wants to convey – promote an idea or express opinions or beliefs toward some topic(s). In such settings, and unlike the machine-translation based efforts (Ritter, Cherry, and Dolan, 2011), the texts should express some disposition of the responder toward the topic, and the text itself should exhibit personal characteristics and beliefs of its author.

In this work we tackle text generation that mimics interpersonal communication in online, social media. Whether to enable effective communication with online users, to support computer-aided interaction, or to interact with the computer in a more personal way, this research is aimed at generating personal and subjective texts. In contrast with technical communication (e.g. procedural reports, legal documents) the research presented here addresses generating texts that carry authors' characteristics such as tone, mood and attitude, for examples, her sentiment towards a certain topic. We concentrate on modeling a human speaker to include characteristics of natural, personal, human language. This model is used as input to a generation pipeline: First, the system compares the user model with an analysis of the context, which determines the response content (*micro planning*); then it is used to augment the generation with additional layers of expression pertaining to the personal attributes or agenda.

Our research is divided into two phases: a proof-of-concept, end-to-end, response generation system. This system is based on *template-based* approaches to generation (Becker, 2002). We use this system to simulate responses in social-media context; the system itself is limited in scope, consisting of a few hand-coded templates, a hand-crafted lexicon, and some world-knowledge representation. To the best of our knowledge, this is the first end-to-end system to generate texts in online context and to target personal communication. Furthermore, as part of the first phase of this research, we establish relevant evaluation methodology for such open domain task by performing a Turing-like test (Turing, 1950) using crowd-sourced judgments via Amazon Mechanical Turk. The result of the first phase a shows that adding world knowledge to a response makes it more relevant. Beyond that, we observed a learning effect between following trials which suggests more variability is required in order to achieve human-like performance.

In the second phase we aim to design a system that can generate a more varied outputs. To this end, we employ a data-driven approach (Konstas and Lapata, 2012a). We first create a relevant dataset that includes news items and user comments from an online news site, and then we parse the data using various kind of grammars, in order to induce syntax-based generation rules. These grammars are also augmented with sentiment markers as in Socher et al. (2013). In this phase we design grammars for effective generation by introducing features that refine the output throughout the derivation of the novel parse trees. While we have not reached the same level of human-likeness as the template-based system exhibits, we are able to empirically show the superiority of relational realizational grammar

on fluency compactness and language model scores, while lexicalized grammars akin to (Collins, 1997) show better human-likeness. We also show empirically that using Topic Models (Papadimitriou et al., 1998; Hofmann, 1999) increase sentence relevance. Our architecture and grammars provide a good starting point for a data-driven generation system that would be more variable, diverse, and relevant for human communication.

Chapter 2

Related Work

The first step in designing a generation system is to determine the communicative goal of the generated utterance. One should then answer the questions of "what to say" and "how to say it", so that the system outputs meet these goals. In our case, we aim to generate texts for personal communication in social context. Such texts, in addition to being relevant, natural and rich, should seem subjective and exhibit personal characteristics, e.g., mood and tone, that are attributed to human writers.

Set in social/interpersonal communication context, our generation system must address and interact with existing content in the media – the generated output should be relevant to previous texts and rely on them for grounding and reference. Furthermore, the way of expression should seem personal and human-like. To meet these goals the texts generated by our system should have some understanding of the context in which they are produced, and convey the human-like and personal characteristics that one would expect from a human responder online.

To answer the questions of "what to say" and "how to say it", our research relies on closely related NLP and AI tasks that allow us to mimic human interaction. As part of the generation flow, textual content is analyzed to retrieve semantic context, author sentiments and possibly other personal aspects. This analysis allows us to infer a state – topics and opinions in the article. Next, when generating a response, the system contrasts, or intersects, the inferred article state, with our model of a responder, and produces a relevant, opinionated and personal text.

In this chapter we present previous work and existing technologies that are relevant to our research goals. In Section 2.1 we discuss natural language understanding; next, in Section 2.2 we address natural language generation (NLG); following that, in Section 2.3 we briefly survey some of the approaches to NLG; we conclude the chapter with a discussion of evaluation of NLG systems in Section 2.4, followed by a short summary in Section 2.5.

2.1 Natural Language Understanding

For the task of generating replies in social communication settings, we first must understand what the discussion is about. Given a natural language text from an online post or a discussion we would like to extract the topics being addressed and the sentiment toward them so that the system can produce a relevant reply.

2.1.1 Understanding Texts

Information Extraction is the task of automatically finding factual information in free text (Piskorski and Yangarber, 2013). Since generation will require some data to address, being able to extract information from text is instrumental to our work. The related task of *Named Entity Recognition* (Nadeau and Sekine, 2007) may

further assist in understanding the text, identifying entities and thus, relating it more clearly to the communicative goal of the generation.

The fields of information extraction (Wimalasuriya and Dou, 2010; Frawley, Piatetsky-Shapiro, and Matheus, 1992) and automatic text summarization (Marcu, 2000; Mani and Maybury, 2001) are widely explored and tackle the task of drawing semantic or structured data from natural language. Given a source text, it will extract the most relevant information from it. The appeal of these work is in trying to normalize or give structure to information from online natural language documents. A similar approach is the semantic parser of Das et al. (2014a). Extracting structured data representation from natural language text sounds appealing but it is not clear whether it can help us in understanding the context for generating a response.

Another approach to understanding text is *Topic Modeling*: a probabilistic generative modeling technique that allows for the discovery of abstract topics over a large body of documents (Papadimitriou et al., 1998; Hofmann, 1999; Blei, Ng, and Jordan, 2003). A trained topic model can then be used to infer the topic(s) mixture of new documents and can be used for our purpose of understanding the discussion.

A leading approach to topic modeling is *Latent Dirichlet Allocation* (LDA, Blei, Ng, and Jordan (2003) and Blei (2012)) which provides the foundations for further research into topic models such as Muti-grain Topic Models (Titov and McDonald, 2008) and Two-Dimensional Topic-Aspect Model (Paul and Girju, 2010). These methods can be leveraged for getting a finer-grain understanding of the topics and be an opening for richer and more refined generated texts.

2.1.2 Understanding Users in Social Media

As is the case with social interactions, human communication is opinionated, and often-time expresses contrastive views toward the topic(s) of the discussion. In order to be able to generate texts in such settings, the generation system must understand not only the topic of the discussion but also what are the views toward these topics. This task can be aided by the ample recent research on opinion mining and sentiment analysis (Pang and Lee, 2008). This large body of work contains the methodologies for analyzing texts and extracting user opinions and characteristics, which will help us define, or refine, the generation system.

Research on characterizing user in social media ranges from large collections of related work on Opinion Mining and Sentiment Analysis (Pang and Lee, 2008) to very specific attempts at understanding one or more aspects of the online interaction. The task of *Sentiment Analysis* uses annotated data and machine learning technique to infer a two-class (Pang and Lee, 2005) or a continuous sentiment-score (Pang, Lee, and Vaithyanathan, 2002) of online reviews. Such work can help us both in the analysis of texts for extracting its sentiment, as well as in corpus analysis when trying to identify language features of online communication that carry such sentiments.

Additional work in NLP incorporates aspects of social relationships to improve the sentiment classification (Tan et al., 2011) or trying to use specific language features of a medium, for examples, tweets in Davidov, Tsur, and Rappoport (2010). Such studies can aid in modeling online user texts. Furthermore, we see various granularity levels at which sentiment is analyzed, such as the aspect-based efforts (Pavlopoulos and Androutsopoulos, 2014). These finer granularity of text analysis capabilities will help in generating finer-grained responses which may seem more human-like.

Another related field is *Emotion recognition*. Emotion recognition is the task of recognizing types of emotions and their strength or intensity in texts (Aman and Szpakowicz, 2007). Emotions are arguably one of the most personal aspects of human communication, and thus, are very relevant to our task. There are studies on this topic (Wu, Chuang, and Lin, 2006; Li et al., 2007) some of which are based explicitly on theories from psychology (Ekman, 1999). For example, SYNESKETCH (Krcadinac et al., 2013) is a software suite that can recognize emotions in texts and is based on Ekman's basic emotions research/categorization.

In the closely related field of *Opinion Mining* (Mishne, 2006), the task is to automatically analyze texts and understand user opinions. A related strand of research is that of *Subjectivity Analysis* (Wilson et al., 2005). In this task, the text are classified as either objective or subjective based on language features. As with sentiment analysis, these works can help improve generation by: (i) getting a better analysis of the discussion and thus being able to generate more relevant and opinionated responses; and (ii) providing another way of verification of the generated texts – applying these tools on our own generated text to verify they are meeting our communicative goals.

Finally, research on computational methods that find out what kind of published utterances are influential, and how they affect linguistic communities (Danescu-Niculescu-Mizil et al., 2009) is complimentary to ours. Such work contributes to studies from sociology and sociolinguistics that aim to delineate the process of generating meaningful responses (e.g., Amabile (1981)). In both cases insights into affective language can be used to augment the generation and produce more convincing and relevant texts.

Dealing with social content, work such as Ritter et al. (2011) may prove important as we are dealing with short and less formal texts.

2.2 Natural Language Generation (NLG)

2.2.1 The Stages of NLG

Natural Language Generation (NLG), the complementary field of natural language processing (NLP), involves the effective generation of texts by computers. In high level, NLG includes three stages (Reiter and Dale, 2000): (i) *macro-planning* which involves content planing and document structuring (what to say); (ii) *micro or sentence planning* which include aggregation, lexicalisation and referring expression generation (how to say); and (iii) *realization* which takes the more abstract sentence planning and creates a surface realization of it – the actual natural language text. Other approaches address the task of NL generation in a more "holistic" way, performing two or more stages jointly. For example Konstas and Lapata (2012b) who combine micro-planning and surface realization in an unsupervised domain-independent fashion, and Zarrieß and Kuhn (2013) that combine referring expression and surface realization as a joint data-driven task.

The *macro-planning* stage is application- or domain-specific. Selecting the relevant information to convey and the high level document structure stems from the communicative goal, the medium of communication and the subject matter itself. It is part of the system high-level design and hence, no specific work is relevant in our case. Still, some general papers about NLG and design of NLG systems are of great use. Reiter, Sripada, and Robertson (2003) discuss knowledge acquisition for

AI/Generation tasks. In this paper the authors presents methodologies for knowledge acquisition such as directly asking experts and corpus analysis and show how to use them correctly to aid in the design of the NLG system.

Micro-planning (Stone et al., 2001) involves both domain specific characteristic such as lexicon, referring expression generation and particular aggregations which can be considered domain specific and must be studied separately. While this is true in general, there is some overlap between micro-planning and realization, such as referring expressions generation, number agreement and aggregations, which can be consider a general task which is often addressed through *realization* libraries. We use such libraries (Gatt and Reiter, 2009) for surface realization tasks such as agreement, sentence structure and general language mechanics. See Section 4.2.2 for how this library was used in our implementation.

It is important to note that a common characteristic of much NLG work is that the generated text is technical, following a concept-to-text approach, a term which broadly refers to the task of automatically producing textual output from non-linguistic input such as databases of records, logical form, and expert system knowledge bases (Reiter and Dale, 2000). Our work is different in that it targets the generation of subjective, opinionated texts that are set in social communication context – a trigger for our system generation is not a database record but an actual interaction or event in the digital media.

2.2.2 Related NLG Tasks

Opinionated NLG is not yet widely explored, but we think that it may become very important as people communicate more and more through virtual mediums. Designing believable agents in computer games (Reilly et al., 1996) is an example of a similar task. More specifically, Strong et al. (2007), discusses an authoring system for "personality rich" characters, which creates a meaningful user interaction in games.

In the same line, research on user interfaces is trying to move away from script-based interaction towards the development of chat bots that attempt natural, human-like interaction (Mori, Jatowt, and Ishizuka, 2003; Feng et al., 2006). However, these chat bots are typically designed to provide an automated one-size-fits-all type of interaction. A related field is that of computerized Personal Digital Assistants – recent releases of such software from Microsoft (Cortana), Apple (Siri) and Google (Google Now) all try to generate relevant and human-like interaction and provide a personalized experience. The personalization is geared toward helping the gadget owner accomplish some tasks but are not subjective or opinionated.

Another interesting field which applies NLG to real-life tasks is that of Augmentative and Alternative Communication (AAC). In AAC (Reiter et al., 2009; Dempster, Alm, and Reiter, 2010) a computerized system aids a handicapped person, who otherwise have difficulties to communicate, to conduct a dialog using the aid of specialized software. The software allows for easily specifying or authoring topics for dialogs and then uses that user data to aid during conversation. While technical in nature (following the concept-to-text approach), Reiter et al. (2009) allows the user to add simple opinion annotation which makes the generated language more personal.

We believe that a framework for generating opinionated and personal texts could be a great contribution to realizing such tasks. Being able to create a more human-like and personal texts could be beneficial to end users, human-computer-interface (HCI) and possibly open the door for more people to interact with other

people and with computers – thus expanding social media outreach to a wider audience.

2.2.3 NLG in Social Media

Unlike NLP research in and of social media context, there is not much work concerning generation in social/interpersonal communication contexts. An exception is a study by Ritter, Cherry, and Dolan (2011), which addresses the generation of responses to natural language tweets in a data-driven setup. It applies a machine-translation approach to response generation, where moods and sentiments already expressed in the past are replicated or reused. A recent study by Hasegawa et al. (2013) modifies Ritter's approach to produce responses that elicit an emotion from the addressee.

These two studies discuss generation efforts that do not address particular topics as a human responder would have done – they can be triggered by any arbitrary text and not only topics of interest. In addition, they do not try to model a user.

In contrast, our goal it to generate new texts based on predefined agenda with topics and sentiments. Here we aim to explicitly model an actual response and generate a genuinely new text. Our take on the task is user centric, generating texts that are personal and explicitly opinionated as a real human interaction would be.

2.3 Approaches to NLG

NLG research has been around for many years (Mann, 1983) and hence, there are many approaches to the various challenges of generating good quality, natural text. The body of work ranges from simple canned-text implementations through more sophisticated template-based or data-driven approaches.

Of interest to our research are works that use template-based approaches (e.g., Becker (2002)) and grammar based ones (e.g. DeVault, Traum, and Artstein (2008) and Konstas and Lapata (2012b)). While the expressive power of the methods is equivalent (Van Deemter, Krahmer, and Theune, 2005) there is variance on the authoring efforts required for each, and the coverage of the generation component (DeVault, Traum, and Artstein, 2008; Narayan, Jr., and Roberts, 2011).

Template-based approaches commonly rely on hand-crafted grammatical constructions that incorporate place-holders for content words. In templates we find some representation of readily available text – whole sentences or phrases – which includes dynamic parts or placeholders which are only realized during runtime, and are changed between invocations of the generation component. The templates can be hand-crafted (Theune et al., 2001) or automatically induced from data (DeVault, Traum, and Artstein, 2008).

In addition for inducing templates, we see other data-driven approaches in generation. Some research follow the traditional separation. For example, Elhadad and Robin (1998) are using a wide-coverage grammar-based surface realizer following content determination and sentence planning tasks (Robin, 1994) that are done separately. In other works, researchers propose a data-driven, empirical methods, which combine the stages into one, making both content-determination and realization decisions in one place (Konstas and Lapata, 2013).

Data-driven grammar-based NLG relies on extracting a grammar from a large corpus of text and then learning how the grammar can be used to realize some functional representation. These methods are usually based on some variation of *Probabilistic Context Free Grammar* (PCFG, Booth and Thompson (1973)). These

settings tend to offer significant advantages over the template-based methods in providing a wider coverage and greater variety. On the other hand, it carries a significant development costs (Busemann and Horacek, 1998).

Data driven approaches to grammar, as presented in DeVault, Traum, and Artstein (2008) mitigate this by using off-the-shelf parsers and learning techniques for realizing text in the respective domain. The approach by Cahill and Genabith (2006) aims to learn functional to structural mapping. Their grammar is geared toward realization of a semantic representation (f-structure) into phrase structure (c-structure), which is in turn used for surface realization. In both cases, the natural language grammar is derived from a corpus. A different approach for grammar-based generation, as in Konstas and Lapata (2012b) and Yuan, Wang, and He (2015), uses a custom defined grammar to find mapping between structure to lingual representation. A common characteristic for both strands of research is the mapping between structure to surface realization. In our research the emphasis is on generation for which the source is also in natural language and not a structured record.

The grammars used in both parsing and generation vary. As seen above, grammars could be custom made (Konstas and Lapata, 2012b; Yuan, Wang, and He, 2015) or derived from corpora (DeVault, Traum, and Artstein, 2008). In our research we will integrate dependency (Tesnière, 1959; Hays, 1964), phrase structure (Chomsky, 1957) and sentiment annotation (Socher et al., 2013) while looking into various existing grammar implementation such as PCFG (Booth and Thompson, 1973), Lexicalized (Collins, 1997) and Relational-Realizational (Tsarfaty and Sima'an, 2008). Each implementation has its own benefits which are discussed in details in Section 5.3.

2.4 Evaluation of Machine-Generated Texts

When evaluating computer generated natural language, there are a few aspects which are important to measure. Outputs have to be natural, fluent and communicate relevant information. Often, the evaluation is task-specific and have to be customized to the research goals. At the same time, all systems share a single ultimate goal: to be conceived as if a real human has generated the text. This brings to mind the famous Turing test (Turing, 1950), a famous test in AI aiming to empirically assess the intelligence of computers.

Evaluation can either be done automatically or rely on human evaluation. In the case of automatic evaluation we see usage of metrics from other fields, for example BLUE (Papineni et al., 2002) or METEOR (Lavie and Agarwal, 2007) from the field of machine translation. Alternatively, measuring coverage, i.e., the generator's ability to re-generate the sources, as in Cahill and Genabith (2006), as indicator of output quality. In recent years, we see also shared tasks (Rus et al., 2011) in NLG and attempts to find a standard evaluation metric for the various NLG tasks (Paris et al., 2007; Foster, 2008). Still there is no specific agreed-upon methodology that fits all tasks. Furthermore, in some instances the automatic metrics correlate well with language quality or human-evaluation score, but do not necessarily give a good measure of content quality (Reiter and Belz, 2009). In other cases, the automatic measures correlate poorly with human evaluation results (Belz and Reiter, 2006).

Alongside efforts to find automated measures, human evaluation is the common approach for evaluating NLG. Human evaluation usually consists of surveys where evaluators are asked to rate or score the output using a predetermine scale and

considering one or more aspects or dimensions of generation (Lester and Porter, 1997). Other methods will show two or more examples to the evaluator and ask her to select the better example. These methodologies can be used to compare two systems and often also involve a mix of both computer generated and real human responses. Having human examples serves both as a quality control check and as means for finding a ceiling rating or a base for comparison (Langner, 2010).

Another human evaluation alternative, task-based approach, aims to measure how suitable the output is for helping a person accomplish a related goal. For example, in Young (1999), generated instructions were tested to measure how well a person can use them to accomplish a task, such as checking out a book from a library or register to classes. Of interesting note for task-based approaches is the conflict between controlled vs. real world (i.e., in context) evaluation (Reiter, 2011). Hence, in a controlled settings, the experiment must be designed in a way that will not affect the results.

In summary, even though human evaluation could be expensive to design and implement, it is often worth the efforts and costs, as the results are more indicative of human-like quality and more faithful to the task at hand. Luckily, with the rise in need for such labor intensive methods, there are more and more platforms available for conducting such studies, as is evident by the vast body of work using such methods (Callison-Burch and Dredze, 2010). Amazon Mechanical Turk is one of the more widely used platforms. In it, a researcher can publish Human Intelligence Tasks (HITs) – tasks which are hard for computers but are relatively easy for a person to solve. Workers can accept a HIT and perform the work, for example, a survey or annotation of data for a predefined fee.

Working in an open domain like ours we would need to find a way to make sure that our responses are relevant and human-like. Also, we will need to verify the personal characteristics of the communication. To verify the output of our own system we will be using the same tools used for analysis of context – for example, using sentiment and subjectivity analysis on our system's outputs to ensure our generated text meets the sentiment goals.

2.5 Summary

In this chapter we provided a brief survey of the fields that are relevant to the main research goals of this thesis.

There is a lot of work on understanding natural language and much of it deals directly (or is immediately applicable) to interpersonal, online, social communication. Relevant research on text understanding includes topic modeling, information extraction, sentiment and subjectivity analysis; these aspects of online traffic are widely explored and well understood.

We have also provided a brief survey of natural language generation. In generation, contemporary work is often geared towards conveying technical information, as means for automation of human–human or computer-aided communication. We surveyed various approaches for natural language text generation. Of note is that these systems usually deal with informative or technical communication; the concept-to-text generation paradigm most often deals with structured data which is "objective" in nature with the exception of some rare examples. We anticipate a rising interest in generating texts in social communication contexts, and in particular, in generating personal and opinionated texts, as we address in this work.

Naturally, the overarching NLG challenge includes the specifics of how to measure the quality of the generated texts. These challenges are amplified in open-domain tasks like ours. The survey on evaluation suggests that automatic measures may be insufficient in our case, and that human evaluation is more promising to our endeavor.

Chapter 3

Research Goals

In this chapter we describe the high level goals as well as concrete research objectives of this thesis. We first outline the context in which it is set – the settings in which we are aiming to achieve the generation goals (Section 3.1). Next, we discuss the concrete objectives of this thesis, define the key questions of the research and outline our approach for answering them (Section 3.2). We conclude with a brief summary (Section 3.3).

3.1 Setting the Stage

Natural language is, above all, a communicative device that we employ to achieve certain goals. In social media, the driving force behind generating responses is an actor with some disposition towards one or more topics. The topics could be a political campaign or candidate, a product, or some abstract idea, which the actor has a motive to promote or demote. In this work we call this goal our user's *agenda*.

In addition to the agenda, there are some other characteristics which are attributed to the actor/user. These characteristics could be static features like stylistic choices, use of voice and other more general traits such as cynicism and humor; or dynamic characteristics such as emotions, which could change the overall realization of the generated texts. Along with the agenda, these features comprise the *user model* and should affect the output of a generation system which aims to imitate the responding human.

Due to the nature of online communication, the content generated by users is usually attached to, or triggered by, some event that is related to the user's agenda. In social media settings, this event is a new *document*, which could be a posting of a news article or a product, a social network update or other online content which the user chooses to respond to. Reacting to such events carries a meaning of its own – the user is opinionated towards the topic and wishes to express her sentiment.

In practice this means that both the *document* and the *user model* form the input to our generation system. We assume that each online document, and each user model, contain (possibly many) topics, each of which is associated with a (positive or negative) sentiment. The generation system should analyze the document to extract these topics and sentiments and infer the disposition of the document's author toward the topic in the document. This disposition should be contrasted with user's agenda to determine the contents of the response.

Similarly to Dale and Reiter (1995), response generation in our work is based on three assumptions, roughly reflecting the Gricean maxims of cooperative interaction (Grice, 1967). Specifically, in this work we aim to generate responses that comply with the following three maxims:

- *Economic* (Maxim of Quantity): Responses should be brief and concise;

- *Relevant* (Maxim of Relation): Responses directly address the documents' content.
- *Opinionated* (Maxim of Quality): Responses express responder's beliefs, sentiments, or dispositions towards the topic(s).

NL generation in itself is a challenging task. Expressing a concept in a natural and fluent way is not trivial for computers, especially when addressing varied topics in an open domain. This research, in addition to the challenges of NLG, introduces a new requirement: making the generated responses relevant and opinionated with respect to a given online *document*. To do so, we must first be able to infer the discussed topics, sentiment and the general communication context.

Once designing such a system for response generation, we also ought to evaluate it. The nature of the domain implies a wide range of topics, and the generated texts can take the form of almost any fluent natural language sentence. Furthermore, the content is likely to change depending on the target medium (e.g., a tweet vs. a comment). As such, there is no way to define a gold standard or a ground-truth for evaluation.

Automated evaluation as was previously proposed for NL generation will not be sufficient in our context (as presented in Section 2.4). New evaluation methodologies should be developed or adapted in order to empirically assess the fluency and relevance of the output of such systems. These methods could rely on human evaluation through crowd-sourcing (e.g. online surveys) and should be carefully devised and executed to yield relevant and non-biased results. As part of our evaluation set-up we also wish to obtain new insights into what aspects of communication make it personal and human-like.

3.2 Research Objectives

Given the general settings describes above, this research aims to address the following research objectives.

3.2.1 Modeling the Responding User

In this research we aim to identify and integrate opinionated and subjective communication aspects into the generation tasks. We would like to model the user itself as an integral part of the generation so that the task is no longer simply concept-to-text but actually user-conception-to-text (or opinion-to-text). Such generation framework should ideally take into account the many aspects of human writing – mood, conviction, attitude and possibly also style of writing and voice. In addition, we would like to consider the associative nature of human memory and its effect on how the response is generated in order to make it more appealing, multi-façade and interesting.

To this end we aim to answer the following questions:

- **Modeling a Responder** – how should we represent a responder? What are the attributes and features that are needed in order to describe a person which interacts in social context? This can be broken down further to several items:
  - **Responder World-Knowledge** – how should we represent the world view of a responder? How can we identify the topics of interest and overall world knowledge the responder has?

  - **Responder Opinions** – how can sentiment, voice, mood and other aspect of a human responder be incorporated in order to aid generation of human-like, subjective and opinionated responses?
  - **Responder "Personality"** – how should personal traits of a responder should be addressed and how can they affect the final response generated?

  In this work we are addressing the first two items, Personality traits and their realization in communication is left for future work.

- **Understanding Context** – what knowledge should be extracted from the triggering content to support an interesting and relevant response generation? How should the context modeling interact with and supplement the responder's model?

- **Generating Human-like responses** – how do the aspects of a human-like response can be addressed technically? How can the additional layer of information can be used to augment existing approaches so that the generated text carries personal dispositions?

3.2.2 Modeling the Response Generation

Following Reiter and Dale (1997), generation should consist of two phases, roughly corresponding to macro and micro planning:

- Macro Planning (below, the *analysis* phase): What are we going to say?

- Micro Planning (below, the *generation* phase): How are we going to say it?

In practice this means that the system should implement an *analysis function* that maps a document to a subjective representation of its content. Each content element may conceivably encompass a topic, its sentiment, its objectivity, its evidentiality, its perceived truthfulness, and so on. Furthermore, additional attributes (such as emotions) can be inferred from the text about the author of the document. In this paper we focus on topic and sentiment, leaving the rest for future research.

Following the analysis, the *generation function* should intersects the content elements in the document with those in the user agenda, and then generate a response based on the content of the intersection. For each non-empty intersection of topics in the document and in the user agendas, our response-generation system aims to generate utterances that are fluent, relevant, and effectively engage readers. These utterances should also be relevant and express the relations between the user model and the document topic dispositions.

Phase 1 - Template-Based Generation

The first phase in our research is the development of a proof-of-concept system. At this stage we aim to develop a simplified, end-to-end, system that performs the task of opinionated content generation. The system will exhibit all of the characteristics defined above. It should produce good results in comparison to human responses. Research by Strong et al. (2007) deals with personality rich generation but in a completely different settings (interactive games). Ritter, Cherry, and Dolan (2011) and Hasegawa et al. (2013) generating language in a similar fields (e.g., tweets), but doing so without explicitly trying to model opinions.

For this stage we plan to develop a system that simulates a responder interacting online. It will rely on hand-crafted resources such as templates, lexicon and knowledge-base, which will be created specifically for this opinionated NLG task. The planned system should lay the foundation and define the architecture of such a solution. Via this first implementation of template-based response generation for a restricted user model we aim to demonstrate the feasibility of the task and design appropriate evaluation measures. Next we will extend the scope of the generation system, and make it entirely data driven.

Phase 2 - Data-Driven Generation

Having established a proof of concept system and an appropriate evaluation methodology, we now turn to scaling up our approach to response generation using a data-driven setup, in order to make the system more robust and diverse.

We do so by employing a grammar-based approach to generation, wherein the grammar for generating responses is induced from a large corpus of online responses. In contrast to standard grammar induction procedures, we aim for a grammar that is also sensitive to lexicalization, selectional restrictions, and sentiment levels.

As often is the case with data-driven generation, there may be exponentially many optional sentences. To address this, we aim to develop a search strategy for finding good sentences. The strategy development should include the mechanism to facilitate efficient search of the generation space as well as a scoring methodology that will allow us to promote the better candidates.

A pre-condition for this phase is the collection of appropriate data for inducing the grammars for generation and training topic models. To our knowledge, such a data-set is not currently readily available in the academia.

3.2.3 Evaluating Generated Texts

As part of the development efforts in the two phases we aim to define and implement an evaluation methodology which is faithful to our task. As stated before, having no standardized evaluation metrics nor a gold standard we will have to rely on human-evaluation as well as on novel use of text analysis tools in order to evaluate our work.

For rating the generated texts and comparing them to human responses we plan to create online surveys that allow users to rate the texts from various configurations of the system. Along with the generation parameters, the system details and external sources (such as tracking survey times and progress), we will use this rating to go beyond simple comparison and try to observe other relevant information that can help improving the generation, by making it more human-like.

In addition to the interactive surveys, we want to be able to use text analysis tools such as sentiment analysis, language models and topic inference in order to aid in evaluation of additional dimensions of this generation task: whether they are topical, opinionated, and relevant.

3.3 Summary

All-in-all we are facing a multidisciplinary challenge: we have to theoretically model user knowledge, personal characteristics and opinions; we have to be able

to extract semantic content and sentiment from online content in order to understand the generation context. Finally, we have to intersect the user model with the context to create relevant text while enhancing traditional concept-to-text approach with personal attributes from the user model. In addition, we have to devise an evaluation methodology that both accounts for the quality of the generated text, and also provides insights into human-like generation of responses.

Chapter 4

Generating Responses: Template-Based Generation

In the first phase of this research we design a proof-of-concept system that simulates the creation of "talkbacks" (user comments in news/content sites). This system uses the template-based approach we discussed in Section 2.3.

In a nutshell, our implementation of the analysis phase uses topic models to infer the topic(s) of an online document, and a simple sentiment analysis system for retrieving the overall sentiment of the document. The retrieved attributes, designated as a *content element*, are then intersected with a predefined user model, consisting of *agendas* as define in the previous chapter, to trigger the template-based generation of a response.

In the generation phase we employ hand-crafted grammatical templates, functions for generating referring expressions, and a small hand-crafted knowledge-base, to generate responses to the triggering document.

In this chapter we describe our model (Section 4.1), analysis and generation architecture (Section 4.2) and a Turing-like novel evaluation procedure to empirically assess our generation results (Section 4.3). We finally provide a more in-depth discussion of the strengths and weaknesses of our system, based on the empirical results (Section 4.4)

4.1 The Model

Let D be a set of documents and let A be a set of user agendas as we formally define shortly. Let S be a set of English sentences over a finite vocabulary $S = \Sigma^*$. Our system implements a function that maps each $\langle document, agenda \rangle$ pair to a natural language response sentence $s \in S$.

$$f_{\text{response}} : D \times A \rightarrow S \tag{4.1}$$

Response generation takes place in two phases, roughly corresponding to macro and micro planning in Reiter and Dale (1997):

- Macro Planning (below, the *analysis* phase): What are we going to say?
- Micro Planning (below, the *generation* phase): How are we going to say it?

The analysis function $c : D \rightarrow C$ maps a document to a subjective representation of its content.[1]

[1] A content element may conceivably encompass a topic, its sentiment, its objectivity, its evidentiality, its perceived truthfulness, and so on. In this paper we focus on topic and sentiment, and leave the rest for future research.

Document sentiment	Agenda sentiment	Response sentiment
positive	positive	positive
positive	negative	negative
negative	negative	positive
negative	positive	negative

TABLE 4.1: The truth table of subjective responses.

The generation function $g : C \times A \rightarrow S$ intersects the content elements in the document and in the user agenda, and generates a response based on the content of the intersection. All in all, our system implements a composition of the analysis and the generation functions:

$$f_{\text{response}}(d, a) = g(c(d), a) = s \tag{4.2}$$

Each content element $c_i \in C$ and agenda item $a \in A$ is composed of a topic, t, associated with a sentiment value $sentiment_t \in [-n..n]$ that signifies the (negative or positive) disposition of the document's author (if $c_i \in C$) or the user's agenda (if $a \in A$) towards the topic.

We assume here that a topic is simply a bag of words from our vocabulary Σ. Thus, we have the following:

$$A, C \subseteq \mathcal{P}(\Sigma) \times [-n..n] \tag{4.3}$$

Following the creation of content element by the analysis function, the system compares the topic(s) in the content element and in the user agenda, and any non-empty intersection of them is used as input to the generation component. The generation component accepts the result of the intersection as input and relies on a template-based grammar and a set of functions for generating referring expressions in order to construct the output.

To make the responses *economic*, we limit the content of a response to one statement about the document or its author, followed by a statement on the relevant topic. To make the response *relevant*, the templates that generate the response make use of topics in the intersection of the document and the agenda. To make the response *opinionated*, the sentiment of the response depends on the (mis)match between the sentiment values for the topic in the document and in the agenda.

Concretely, the response is positive if the sentiments for the topic in the document and agenda are the same (both positive or both negative) and it is negative otherwise. This is effectively captured in Table 4.1.

We suggest two variants of the generation function g. The basic variant implements the baseline function defined above:

$$g_{\text{base}}(c, a) = s$$

$$c \in C, a \in A, s \in \Sigma^*$$

For the other variant we define a knowledge base (KB) as a directed graph in which words $w \in \Sigma$ from the topic models correspond to nodes in the graph, and relations $r \in R$ between the words are predicates that hold in the real world. Our second generation function now becomes:

$$g_{\text{kb}}(c, a, \boldsymbol{KB}) = s$$

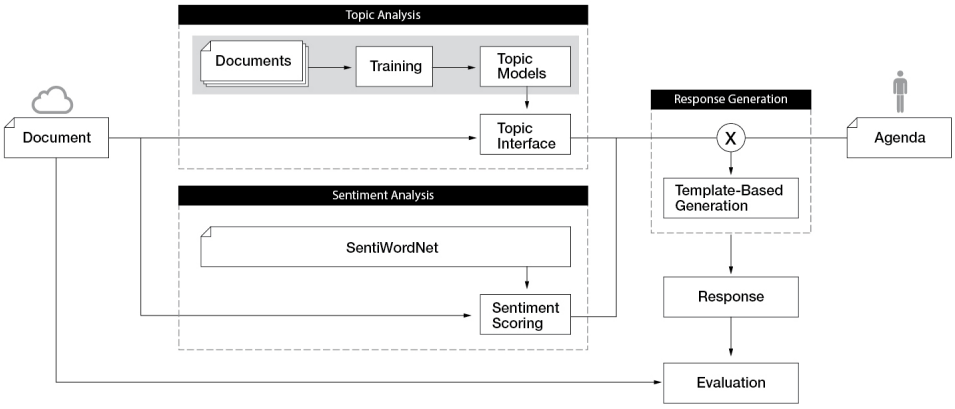


FIGURE 4.1: The system architecture from a bird's eye view. Components with gray background are executed offline.

$$KB \subseteq \{(w_i, r, w_j) | w_i, w_j \in \Sigma, r \in R\}$$

with $c \in C, a \in A, s \in \Sigma^*$ as defined in g_{base} above.

4.2 The Architecture

The system architecture from a bird's eye view is presented in Figure 4.1. In a nutshell, a document enters the analysis phase, where topic inference and sentiment scoring take place, resulting in ⟨topic, sentiment⟩-pairs. During the subsequent generation phase, these are intersected with the ⟨topic, sentiment⟩-pairs in the user agenda. This intersection, possibly augmented with a knowledge graph, forms the input for a template-based generation component.

4.2.1 Analysis phase

For the task of inferring the topics of the document we use topic modeling: a probabilistic generative modeling technique that allows for the discovery of abstract topics over a large body of documents (Papadimitriou et al., 1998; Hofmann, 1999; Blei, Ng, and Jordan, 2003). Specifically, we use topic modeling based on *Latent Dirichlet Allocation* (LDA) (Blei, Ng, and Jordan, 2003; Blei, 2012). A topic in this context extends our "bag of words" definition with a probability distribution over words. A topic model provides a probability distribution over topics for each document, and a for each topic, a probability distribution over the observed words. The result of training a Topic Model is a list of vectors of fixed length reflecting words prevalence in the document. Each vector represents a topic t and each element (a word) in the vector has a probability of having been generated by the topic. Given a new document and a trained model, the inference method provides a weighted mix of topics for that document, where each topic is represented as a vector containing keywords associated with probabilities. For training the topic model and inferring the topics in new documents we use *Gensim* (Rehurek and Sojka, 2010), a fast and easy-to-use implementation of LDA.

Next, we wish to infer the sentiment that is expressed in the text with relation to the topic(s) identified in the document. We use the semantic/lexical method as implemented in Kathuria (2012). We rely on a WSD sentiment classifier that uses

the SentiWordNet (Baccianella, Esuli, and Sebastiani, 2010) database and calculates the positivity and negativity scores of a document based on the positivity and negativity of individual words. The result of the sentiment analysis is a pair of values, indicating the positive and negative sentiments of the document-based scores for individual words. We use the larger of these two values as the sentiment value for the whole document.[2]

4.2.2 Generation phase

Our generation function first intersects the set of topics in the document and the set of topics in the agenda in order to discover relevant topics to which the system would generate responses. A response may in principle integrate content from a range of topics in the topic model distribution, but, for the sake of generating concise responses, in the current implementation we focus on the single most prevalent, topic. We pick the highest scoring word of the highest scoring topic, and intersect it with topics in the agenda. The system generates a response based on the identified topic, the sentiment for the topic in the document, and the sentiment for that topic in the user agenda.

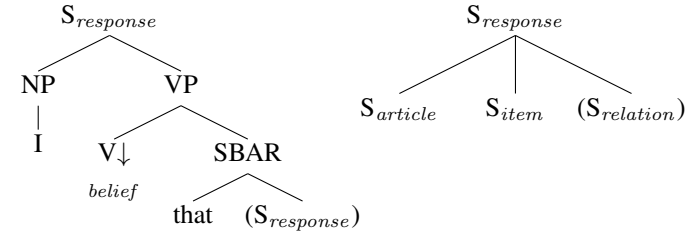
The generation component relies on a template-based approach similar to Reiter and Dale (1997) and Van Deemter, Krahmer, and Theune (2005). Templates are essentially subtrees with leaves that are place-holders for other templates or for functions generating referring expressions (Theune et al., 2001). These functions receive (relevant parts of) the input and emit the sequence of fine-grained part-of-speech (POS) tags that realizes the relevant referring expression. The POS tags in the resulting sequences are ultimately place holders for words from a lexicon, Σ. In order to generate a variety of expression forms — nouns, adjectives and verbs — these items are selected randomly from a fine-grained lexicon we defined. The sentiment (positive or negative) is expressed in a similar fashion via templates and randomly selected lexical entries for the POS slots, after calculating the overall sentiment for the intersection as stated above. Our generation implementation is based on SimpleNLG (Gatt and Reiter, 2009) which is a surface realizer API that allows us to create the desired templates and functions, and aggregates content into coherent sentences. The templates and functions that we defined are depicted in Figure 4.2.

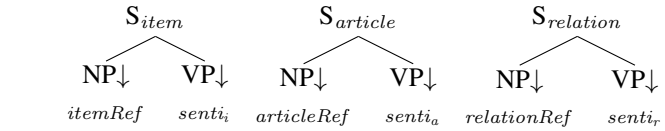
In addition, we handcrafted a simple knowledge graph (termed here KB) containing the words in a set of pre-defined user agendas. Table 4.2 shows a snippet of the constructed knowledge graph. The knowledge graph can be used to expand the response in the following fashion: The topic of the response is a node in the KB. We randomly select one of its outgoing edges for creating a related statement that has the target node of this relation as its subject. The related sentence generation uses the same template-based mechanism as before. In principle, this process may be repeated any number of times and express larger parts of the KB. Here we only add one single knowledge-base relation per response, to keep the responses concise.

4.3 Evaluation

We set out to evaluate how computer-generated responses compare to human responses in their perceived *human-likeness* and *relevance*. More in particular, we

[2] Clearly, this is a simplifying assumption. We discuss this assumption further in Section 4.4.





articleRef	← ExpressArticle(...)
itemRef	← ExpressItem(...)
relationRef	← ExpressRelation(...)
sentiment$_a$	← ExpressArticleSentiment(...)
sentiment$_i$	← ExpressItemSentiment(...)
sentiment$_r$	← ExpressRelationSentiment(...)
belief	← ExpressBelief(...)

FIGURE 4.2: Template-based response generation. The templates are on the left. The Express* functions on the right use regular expressions over the arguments and vocabulary items from a predefined lexicon.

Source	Relation	Target
Apple	CompetesWith	Samsung
Apple	CompetesWith	Google
Apple	Creates	iOS

TABLE 4.2: A knowledge Graph Snippet.

Sent.	KB	Response
−8	No	Android is horrendous so I think that the writer is completely correct!!!
	Yes	Apple is horrendous so I feel that the author is not really right!!! iOS is horrendous as well.
−4	No	I think that the writer is mistaken because apple actually is unexceptional.
	Yes	I think that the author is wrong because Nokia is mediocre. Apple on the other hand is pretty good ...
0	No	The text is accurate. Apple is okay.
	Yes	Galaxy is okay so I think that the content is accurate. All-in-all samsung makes fantastic gadgets.
4	No	Android is pretty good so I feel that the author is right.
	Yes	Nokia is nice. The article is precise. Samsung on the other hand is fabulous...
8	No	Galaxy is great!!! The text is completely precise.
	Yes	Galaxy is awesome!!! The author is not completely correct. In fact I think that samsung makes awesome products.

TABLE 4.3: Responses generated by the system with or without a knowledge-base (KB), with different sentiment levels.

compare different system variants in order to investigate what makes responses seem more human-like or relevant.

4.3.1 Materials

Our empirical evaluation is restricted to topics related to mobile telephones, specifically, Apple's iPhone and devices based on the Android operating system. We collected 300 articles from leading technology sites in the domain to train the topic models on, settling on 10 topics. Next, we generated a set of user agendas referring to the same 10 topics. Each agenda is represented by a single keyword from a topic model distribution and a sentiment value $sentiment_t \in \{-8, -4, 0, 4, 8\}$. Finally, we selected 10 new articles from similar sites and generated a pool of 1000 responses for each, comprising 100 unique responses for each combination of $sentiment_t$ and system variant (i.e., with or without a knowledge base). Table 4.3 presents an example response for each such combination. In addition, we randomly collected 5 to 10 real, short or medium-length, online human responses for each article.

4.3.2 Surveys

We collected evaluation data via two online surveys on Amazon Mechanical Turk (www.mturk.com). In Survey 1, participants judged whether responses to articles were written by human or computer, akin to (a simplified version of) the Turing test (Turing, 1950). In Survey 2, responses were rated on their relevance to the article, in effect testing whether they abide by the Gricean Maxim of Relation. This is comparable to the study by Ritter, Cherry, and Dolan (2011) where people judged which of two responses was 'best'.

Each survey comprises 10 randomly ordered trials, corresponding to the 10 selected articles. First, the participant was presented with a snippet from the article. When clicking a button, the text was removed and its presentation duration recorded. Next, a multiple-choice question asked about the snippet's topic. Data on a trial was discarded from analysis if the participant answered incorrectly or if the snippet was presented for less than 10 msec per character; we took these to be cases where the snippet was not properly read. Next, the participant was shown a randomly ordered list of responses to the article.

In Survey 1, four responses were presented for each article: three randomly selected from the pool of human responses to that article and one generated by

Response Type	Mean and CI
Human	3.33 ± 0.08
Computer (all)	4.49 ± 0.15
Computer ($-$KB)	4.66 ± 0.20
Computer ($+$KB)	4.32 ± 0.22

TABLE 4.4: Mean and 95% confidence interval of computer-likeness rating per response category. $\pm$KB indicates whether g_{base} or g_{kb} was used.

Factor	b	t	$P(b < 0)$
(intercept)	3.590		
IS_COMP	0.193	2.11	0.015
POS	0.069	4.76	0.000
IS_COMP $\times$ POS	0.085	6.27	0.000

TABLE 4.5: Computer-likeness rating regression results, comparing human to computer responses.

our system. The task was to categorize each response on a 7-point scale with labels 'Certainly human/computer', 'Probably human/computer', 'Maybe human/computer' and 'Unsure'. In Survey 2, five responses were presented: three human responses and two computer-generated. The task was to rate the responses' relevance on a 7-point scale labeled 'Completely (not) relevant', 'Mostly (not) relevant', 'Somewhat (not) relevant', and 'Unsure'. As a control condition, one of the human responses and one of the computer responses were actually taken from another article than the one just presented. In both surveys, the computer-generated responses presented to each participant were balanced across sentiment levels and generation functions (g_{base} and g_{kb}). After completing the 10 trials, participants provided basic demographic information, including native language. Data from non-native English speakers was discarded. Surveys 1 and 2 were completed by 62 and 60 native speakers, respectively.

4.3.3 Analysis and Results

Survey 1: Computer-Likeness Rating.

Table 4.4 shows the mean 'computer-likeness'-ratings from 1 ('Certainly human') to 7 ('Certainly computer') for each response category. Clearly, the human responses are rated as more human-like than the computer-generated ones: our model did not generally mislead the participants. This may be due to the template-based response structure: over the course of the survey, human raters are likely to notice this structure and infer that such responses are computer-generated. To investigate whether such learning indeed occurs, a linear mixed-effects model was fitted, with predictor variables IS_COMP ($+1$:computer-generated, -1:human responses), POS (position of the trial in the survey, 0 to 9), and the interaction between the two. Table 4.5 presents, for each factor in the regression analysis, the coefficient b and its t-statistic. The coefficient equals the increase in computer-likeness rating for each unit increase in the predictor variable. The t-statistic is indicative of how much variance in the ratings is accounted for by the predictor. We also obtained a probability distribution over each coefficient by Markov Chain Monte Carlo sampling

Factor	b	t	$P(b < 0)$
(intercept)	4.022		
KB	−0.240	−2.13	0.987
POS	0.144	5.82	0.000
SENT	0.035	2.98	0.002
abs(SENT)	−0.041	−1.97	0.967
KB × POS	0.023	1.03	0.121

TABLE 4.6: Computer-likeness rating regression results, comparing systems with and without KB.

using the R package lme4 version 0.99 (Bates, 2005). From each coefficient's distribution, we estimate the posterior probability that b is negative, which quantifies the reliability of the effect.

The positive b value for POS shows that responses drift towards the 'computer'-end of the scale. More importantly, a positive interaction with IS_COMP indicates that the difference between human and computer responses becomes more noticeable as the survey progresses — the participants did learn to identify computer-generated responses. However, the positive coefficient for IS_COMP means that even at the very first trial, computer responses are considered to be more computer-like than human responses.

Factors Affecting Human-Likeness. Our finding that the identifiability of computer-generated responses cannot be fully attributed to their repetitiveness, raises the question: What makes a such a response more human-like? The results provide several insights into this matter.

First, the mean scores in Table 4.4 suggest that including a knowledge base increases the responses' human-likeness. To further investigate this, we performed a separate regression analysis, using only the data on computer-generated responses. This analysis also included predictors KB (+1: knowledge base included, −1: otherwise), SENT ($sentiment_t$, from −8 to +8), absolute value of SENT, and the interaction between KB and POS. As can be seen in Table 4.6, there is no reliable interaction between KB and POS: the effect of including the KB on the human-likeness of responses remained constant over the course of the survey.

Furthermore, we see evidence that responses with a more positive sentiment are considered more computer-like. The (only weakly reliable) negative effect of the absolute value of sentiment suggests that more extreme sentiments are considered more human-like. Apparently, people count on computer responses to be mildly positive, whereas human responses are expected to be more extreme, and extremely negative in particular.

Survey 2: Relevance Rating.

The mean relevance scores in Table 4.7 reveal that a response is rated as more relevant to a snippet if it was actually a response to that snippet, rather than to a different snippet. This reinforces our design choice to include input items referring specifically to the topic and sentiment of the author. However, human responses are considered more relevant than the computer-generated ones. This is confirmed by a reliably negative regression coefficient for IS_COMP (see regression results in Table 4.8).

Response Type	Source	Mean and CI
Human	this	4.85 ± 0.11
	other	3.56 ± 0.18
Computer (all)	this	4.52 ± 0.16
	other	2.52 ± 0.15
Computer ($-$KB)	this	4.53 ± 0.23
	other	2.46 ± 0.21
Computer ($+$KB)	this	4.51 ± 0.23
	other	2.58 ± 0.22

TABLE 4.7: Mean and 95% confidence interval of relevance rating per response category. 'Source' indicates whether the response is from the presented text snippet or a random other snippet. $\pm$KB indicates whether g_{base} or g_{kb} was used.

Factor	b	t	$P(b < 0)$
(intercept)	3.861		
IS_COMP	-0.339	-7.10	1.000
SOURCE	0.824	16.80	0.000
IS_COMP $\times$ PRES	0.179	5.03	0.000

TABLE 4.8: Relevance ratings regression results, comparing human to computer responses.

The analysis included the binary factor SOURCE ($+1$ if the response came from the presented snippet, -1 if it came from a random article). We see a positive interaction between SOURCE and IS_COMP, indicating that presenting a response from a random article is more detrimental to relevance of computer-generated responses than that of the human responses. This is not surprising, as the computer-generated responses (unlike the human responses) always includes the article's topic.

When analyzing only data on computer-generated responses, and including predictors for agenda sentiment and for presence of the knowledge base, we see that including the KB does not affect response relevance (see Table 4.9). Also, there is no interaction between KB and SOURCE, that is, the effect of presenting a response from a different article does not differ between the models with and without the knowledge base. Possibly, responses are considered as more relevant if they have more positive sentiment, but the evidence for this is fairly weak.

Factor	b	t	$P(b < 0)$
(intercept)	3.603		
KB	0.026	0.49	0.322
SOURCE	1.003	15.90	0.000
SENT	0.023	1.94	0.029
abs(SENT)	-0.017	-0.93	0.819
KB $\times$ SOURCE	-0.032	-0.61	0.731

TABLE 4.9: Relevance ratings regression results, comparing systems with and without KB.

4.4 Discussion and Future Work

Alongside the vast amount of research on sentiment and topic analysis, and in contrast to most generation tasks that uses artificial or pre-defined structured input, we implemented a end-to-end system that completes the full cycle from natural language analysis to natural language generation with applications in social media and automated interaction in real-world settings.

The only two other studies on response generation in social media we know of are Ritter, Cherry, and Dolan (2011) and Hasegawa et al. (2013). Ritter's and Hasegawa's approaches differ from ours in their objective and their approach to generation. Specifically, Ritter's approach is based on machine translation, creating responses by directly re-using previous content. Their data-driven approach generates relevant, but not explicitly opinionated responses. In addition, both Ritter's and Hasegawa's systems respond to tweets, while our system analyzes and responds to complete articles. Hasegawa's approach is closer to ours in that it generates responses that are intended to elicit a specific emotion from the addressee. However, it still differs considerably in settings (dialogues versus online posting) and in the goal itself (eliciting emotion versus expressing opinion). Thus, we see these studies as complementary to ours in the realm of response generation in social media.

Topic model prediction provides a rich form of input (probability distribution over words, according to a mix of topics), from which the current system isolates a top-ranked keyword that represents the most prominent topic. Clearly, the generation can refer to multiple topics, or integrate multiple keywords from certain topics. Using the topic model more heavily could improve the grounding, and subsequently, the relevance, of the generated responses. Furthermore, topic model word probabilities could be used in grammar-based approach to drive parts of the generation, a notion that will be explore in the second part of the research.

Likewise, our sentiment analysis component uses a general-purpose implementation that calculates a single sentiment for the entire document. In a data-driven approach, this could possibly be expanded to use a more fine-grained sentiment analysis (e.g. phrase-structure attached sentiment as in Socher et al. (2013)).

The syntactic and semantic means of expression that we used are based on bare bone templates and fine-grained POS tags (Theune et al., 2001). These may potentially be expanded with different ways to express subject/object relations, relations between phrases, polarity of sentences, and so on. Additional approaches to generation can factor in such aspects, e.g., the template-based methods in Becker (2002) and Narayan, Jr., and Roberts (2011), or grammar based methods, as in DeVault, Traum, and Artstein (2008). Using more sophisticated generation methods with a rich grammatical backbone may help to overcome the sensitivity to computer-generated response patterns as acquired by our human raters over time.

4.5 Conclusions

We presented a system for generating responses that are directly tied to responders' agendas and document content. To the best of our knowledge, this is the first system to generate subjective responses directly reflecting users' agendas. Our response generation architecture provides an easy-to-use and easy-to-extend solution encompassing a range of NLP and NLG techniques. We evaluated both the human-likeness and the relevance of the generated content, thereby empirically

quantifying the efficacy of computer-generated responses compared head-to-head against human responses.

Generating concise, relevant, and opinionated responses that are also human-like is hard — it requires the integration of text-understanding and sentiment analysis, and it is also contingent on the expression of the agents' prior knowledge, reasons and motives. We suggest our architecture and evaluation method as a baseline for future research on generated content that would effectively pass a Turing-like test, and successfully convince humans of the authenticity of generated responses.[3]

[3]The code, training and experimental data (computer and human responses) and analysis scripts for this chapter are publicly available via http://tomercagan.com/onlg.

Chapter 5

Generating Responses: Data-Driven Generation

The limited ability of template-based generation to produce a large diversity of responses and the diminishing human-likeness scores as the survey progressed were the main conclusions drawn from the first phase of our research. As already noted by others, the expressive power of templates is as good as that of other methods (Van Deemter, Krahmer, and Theune, 2005), however, they suffer from poorer coverage (DeVault, Traum, and Artstein, 2008) and high associated authoring costs (Narayan, Jr., and Roberts, 2011).

With these observations in mind, we now turn to the second phase of our work, where we set out to explore a data-driven implementation for the opinionated response generation task. Our goal is to deliver an end-to-end system that is more flexible and robust, in the hope of getting good human-likeness and relevance scores, in a more open framework, which could work across domains and with less required coding and resource authoring efforts.

To the best of our knowledge, this is the first use of grammar-based generation methods for generating opinionated responses. To achieve this, we have to first create a relevant data set, and then study the ways in which grammar-based generation may be guided by opinionated or personal features.

Our approach to data-driven generation combines micro-planning and realization as in Konstas and Lapata (2013). In this thesis, they induce transition probabilities for predefine grammar which is used to drive both micro-planning and realzation at once. Following a similar approach, the result of our data-driven learning is a *decoder/generator*, that, given new context and an instance of user model, generates utterance(s) which take into account the responder model, emitting grammatical language structures and appropriate lexical choices.

In the following sections we survey the formal model (5.1) and overall architecture (5.2), and then discuss in details the grammars we induced for the generation and how we acquired them from data (5.3). We finally evaluate these grammars for generation empirically, via a mix of human-based and automatic methods (5.4), and then we finalize and conclude in Section 5.5.

5.1 The Model

Similar to the model presented in Chapter 4, we define an online document, d, and an analysis function c, to extract content elements from the document. The content is intersected with the agenda, a, defined in our user model, for matching topics and sentiment.

Taking a data-driven approach, we aim to learn the structure of opinionated responses. Given the scheme above and a data-set of online documents and corresponding comments, we want to extract a grammar, decorated with different linguistic constructs such as phrase-structure, dependency and sentiment. A high level overview of this process is depicted in Figure 5.1.

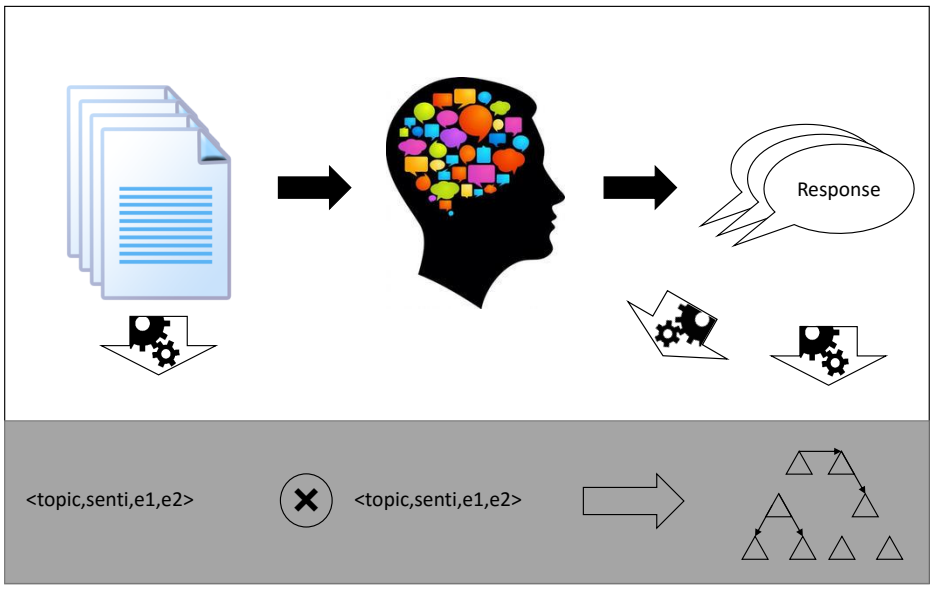


FIGURE 5.1: Overview of the learning process. Context is extracted from source documents. Real responses are used to extract responder attributes and learn grammar rules, weights, and a language model.

More formally, in data-driven, grammar-based generation settings, we have the following overall objective function:

$$F(a, d) = \underset{w,t}{\operatorname{argmax}}\, P(w, t|\phi(a \otimes c(d))) \tag{5.1}$$

where t is a derivation tree, and w is the yield of that tree. ϕ is a dimensionality reduction function which returns the relevant features from the intersection of a and $c(d)$. In our case, this would be sentiment and topics or lexical items. Considering some induced grammar, $G = (N, \Sigma, R, S)$ where:

- N is a finite set of non-terminal symbols.
- Σ is a finite set of terminal symbols.
- R is a finite set of transition rules of the form $X \rightarrow Y_1\ Y_2$.
- $S \in N$ is a distinguished start symbol.

we can expand the objective function in Equation 5.1 to a distinct set of decisions:

$$\begin{aligned} P(w, t|\phi(a \otimes c(d))) = & P(root|\phi(a \otimes c(d))) \quad \times & \text{(5.2a)} \\ & P(t|root, \phi(a \otimes c(d))) \quad \times & \text{(5.2b)} \\ & P(w|t, \phi(a \otimes c(d))) & \text{(5.2c)} \end{aligned}$$

In words, we first select a start rule, $root$, based on the content of the intersection, $(a \otimes c(d))$ (5.2a), we then select a derivation tree, t given the $root$ and the content (5.2b), and finally, we select the surface realization, w, based on the generated tree (5.2c).

Relating this method to the concept-to-text model presented in Konstas and Lapata (2013), we can draw similarities between our chain of decisions to the set of decisions executed by their decoder: where Konstas and Lapata select records from a database, we select starting rule(s); where their model selects fields to include in the response, it may be seen as equivalent to the CFG rules that derive the tree in our model. Finally, selecting words for realizing the fields in their model is similar to selecting non-terminal symbols, that is, words, for surface realization of the response in our model.

Konstas and Lapata (2013) demonstrate data-driven generation by re-interpreting a semi-hidden Markov model that find correspondence between the decisions in each level of the hierarchy as CFG rewrite rules. We perform a similar process by using a PCFG-like grammar. Using independence assumptions, and the learned emission probabilities, Equation 5.2 can be re-written as a chain of local decisions:

$$P(w,t|\phi(a \otimes c(d))) \approx P(root|\phi(a \otimes c(d))) \quad \times \tag{5.3a}$$

$$\prod_{rule \in t} P(rule|root, \phi(a \otimes c(d))) \quad \times \tag{5.3b}$$

$$\prod_{i=0}^{k} P(w_i|t, \phi(a \otimes c(d))) \tag{5.3c}$$

In Equation 5.3a we select a rule of the form $TOP \rightarrow Cat$, where Cat is a category encompassing a phrase-structure category, possibly augmented with lexical information and sentiment relevant to the response about to be generated. In 5.3b the response tree is generated by continually selecting derivation rules from the induced grammars. Each derivation rule may make syntactic, lexical, and sentiment choices while advancing the frontier. We introduce three types of grammars (Section 5.3) and empirically compare them (Section 5.4). Note that the derivation of the tree (5.3b) is unbounded as the tree can be expanded to an arbitrary depth. In practice we limit the tree depth to 13. This gives a bound to i in the surface realization (5.3c). See Section 5.2 for further implementation details of this matter.

5.2 The Architecture

To follow the definition of our model and the objective function defined in the previous section, we designed a data-driven, grammar-based generation pipeline. For realizing the various decisions objectives in Equation 5.3 we need to use a variety on PCFG (Cahill and Genabith, 2006), which in addition to the components described above, also includes a transition probability component for each rule in R.

The components in our pipeline includes corpus of relevant texts, a parser for processing the text and inducing grammar and emission probabilities, and an generator/decoder that uses the grammar, and given a document and a user model, generates opinionated sentences. A bird's-eye view of this pipeline is depicted in Figure 5.2.

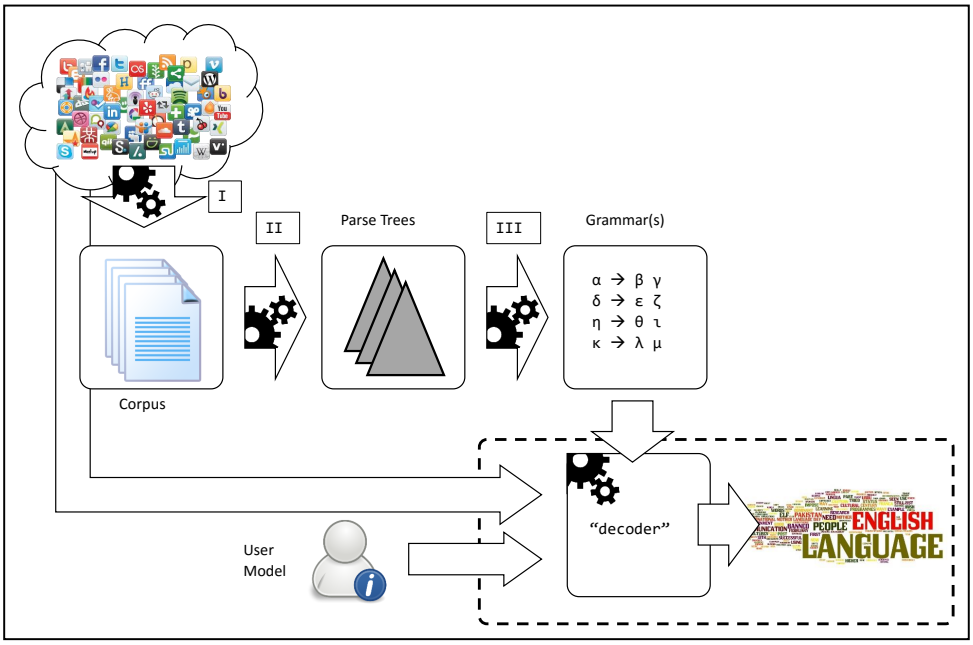


FIGURE 5.2: Data-driven, grammar-based generation system architecture overview. The process consists of collecting a corpus (I), parsing and annotation (II), and grammar induction (III). The induced grammar along with the user model and new online document are used during the generation. The dashed box encompasses the online generation flow.

A preliminary step in our pipeline is the collection of a relevant corpus which contains pairs of document and corresponding user comments. The documents from the corpus are used for training a topic model which is then used during generation for topic inference, similar to the analysis step in the template-based approach in Section 4.2. The user comments are used for grammar induction and for calculating emission probabilities, and are described in more details in Section 5.3. We collected such a corpus and present it in Section 5.4.1.

To realize the goal of generation an opinionated texts, we decided to jointly model opinion, structure and lexical decision in our pipeline by inducing grammar which is annotated with lexical choice and sentiment in addition to phrase-structure and dependency information. The high level grammar induction is described in Section 5.2.1.

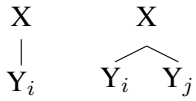
Finally, we defined a generator which carries the generation as described in our model (Equation 5.3). As is common in tasks where the generation space is exponential, the generator uses an efficient search strategy to find and further develop the best candidates. In addition, it over generates and re-rank candidates. These steps are described in Section 5.2.2.

In contrast to common concept-to-text approaches, our data-driven approach for text generation is set in an unstructured context: the generation of interpersonal communication based on online content. Therefore, some preliminary processing of the corpus is required, in order to obtain the main content elements expressed and the grammatical constructs that express them.

5.2.1 Grammar Extraction

We follow a common methodology for inducing a PCFG from our data-set: first, user comments from the corpus are parsed using an off-the-shelf parser, using different kinds of grammar we devise, to yield a parse tree for each sentence in each comment; next, the parse trees are traversed, counting all occurrences of each rule; the counts are then used to estimate the occurrence probabilities of each rule. In our case the parse trees contain unary (root with one child) and binary (root with two children) rules.

For the rest of this discussion, we will use the following notation for parse-trees and corresponding derivation rules:



These depth-1 derivations are equivalent to CFG rules, which we indicate as:

$$X \rightarrow Y_i \qquad X \rightarrow Y_i \;\; Y_j$$

More generally, we use $\alpha \rightarrow \beta$ to designate either unary or binary grammar rule that corresponds to depth-1 derivations.

As for deriving rule probabilities, the trees are traversed to retrieve and count the occurrences of each α and $\alpha \rightarrow \beta$ and apply simple maximum likelihood estimation (MLE), which takes the following form:

$$P(\alpha \rightarrow \beta) = \frac{count(\alpha \rightarrow \beta)}{count(\alpha)}$$

In the rest of this discussion, unless stated otherwise, we have used version 3.5.2 of the Stanford Parser (Klein and Manning, 2003) through the StanfordCoreNLP interface and with the default settings. For dependency parsing we used the grammatical structure functionality directly with a SemanticHeadFinder, and not through the StanfordCoreNLP pipeline.

The main advantage of using the Stanford libraries in the context of this thesis, is the availability of sentiment annotation at the phrase-structure level. Based on the work of (Socher et al., 2013) the parser emit a sentiment classification for each node in the phrase-structure tree. The sentiment annotation is realized through a Recursive Neural Tensor Network which is trained over a very large annotated tree-bank. In a sense, this implementation captures sentiment compositionality, clearly outperform any similar systems and is the only model that can accurately capture the effect of contrastive conjunctions as well as negation and its scope at various tree levels for both positive and negative phrases. An example of sentiment annotated constituency node can be seen in Figure 5.5(a).

In addition, the Stanford libraries are simple to get started with, requiring minimal tinkering to get up and running and give all the required functionality in one set of libraries which are flexible enough for future customization.

5.2.2 Grammar-Based Generation

Generation with grammar, at its simplest form, consists of choice of derivation rules to apply at the frontier, and gradually expending the tree derived by these rules. In a top-down approach, starting with a frontier that includes only the selected root, the tree is expanded continually by substituting non-terminals at the left-hand-side

with their daughters on the right hand side, until no more non-terminals exist. The pseudo code for generation is listed in Algorithm 1

Algorithm 1 Basic Generation

```
1:  procedure GENERATE
2:      root = selectRoot()                          ▷ select start rule
3:      processingList.add(root)                     ▷ add to processing list
4:      while not processingList.isEmpty() do        ▷ while there are nodes to process
5:          node = processingList.get()              ▷ get the next node
6:          for all child in node.children do        ▷ process each child of the node
7:              if not child.isTerminal() then       ▷ only expand non-terminals
8:                  rule = getRule(child)            ▷ get rule for the child
9:                  child.addChildren(rule)          ▷ and add as children
10:                 processingList.add(child)        ▷ add to list for processing
11:             end if
12:         end for
13:     end while
14:     return root                                  ▷ root now has derivation tree
15: end procedure
```

In view of this algorithm, several implementation notes are due. First, the implementation of *processingList* as a queue or a stack will determine the expansion/traversal order of the tree in breadth- or depth-first respectively. Here we used the "generic" terms *list*, *add* and *get* and explore the various alternatives in our implementation. Next, implementation of a node here is a simple, labeled, element with a list of children of the same type, and a function to determine if it is a terminal node or not. Selecting the root in *selectRoot* limits the possible rules available in expansion of the tree and hence it is an important decision. This decision may determine the topic of the responses, the desired sentiment, and crucial lexical choices (such as the head of the phrase). We discuss this when evaluating responses relevance in Section 5.4.2. Finally, Selecting the rules in *getRule* for expanding the tree is at the heart of this research as these rules determine the actual content of the response. We discuss this aspect further in our survey of the induced grammars in Section 5.3.

Over-Generation and Re-ranking

The generation procedure described in Algorithm 1 yields one sentence for any given root rule. In template-based or canned-text approaches one has a guarantee that the sentence is grammatical and, due to usage of a predefined lexicon, also relevant with correct lexical choices. The same does not always holds true for grammar-based methods due to the limiting independence assumptions when applying CFG rules in the derivation of the tree. Selecting rules for expansion based on local (context-free) nodes may end up in sentences which are not grammatical, due to an incoherent phrase structure, or are non-relevant in terms of content, due to wrong lexical choices.

To overcome this inherent limitation of grammar-based approaches it is common to generate multiple candidates for a specific instance of the task and employ some re-ranking or re-selection process through the generation steps (or after the

generation has been completed) and choose from the many candidates that are being created. Re-ranking during the generation is usually done in bottom-up generators, as in chart-generation (Kay, 1996; Shieber et al., 1989). When using over-generation and re-ranking, the tree is being built from the bottom and expanded up using some heuristics which rely on various features. For example, a language model and alignment features in Konstas and Lapata (2012a).

Unlike Konstas and Lapata (2012a), which starts from a structured representation and can possibly limit the initial options explored in the leaves, we map an unrestricted user-agenda to an unstructured and open domain, hence, cannot handily create a bottom-up generator. While chart generation (Haruno, Den, and Matsumoto, 1996) is an option, we opt for top-down generation, which resembles the top-down perspective of the template-based responses in Chapter 4. Such generators can be viewed as having two phases. In the first, initial relevant rules are selected and expanded to create many possible derivation trees (or a *forest*). The second phase consists of searching that forest to retrieve the best trees.

Our generation algorithm then resembles the pseudo-code in Algorithm 2. Note that in this approach, we define a generation node with multiple options, each of which is equivalent to a derivation rule available for expansion within that node. These options are developed further in line 7 of the algorithm. Additional options are added for each node in line 11.

Algorithm 2 Multi-Option Generation

```
1:  procedure GENERATEMULTI
2:      root = selectRoot()                                  ▷ select start rule
3:      processingList.add(root)                             ▷ add to processing list
4:      while not processingList.isEmpty() do  ▷ while there are nodes to process
5:          node = processingList.get()                      ▷ get the next node
6:          for all option in node.options do    ▷ process each optional rule in the
    node
7:              for all child in option.children do          ▷ process each child of the
    optional derivation
8:                  if not child.isTerminal() then           ▷ only expand non-terminals
9:                      rule = getRules(child)               ▷ get rules for the child
10:                     for all rule in rules do
11:                         child.addOption(rule)            ▷ (and an optional rule)
12:                     end for
13:                     processingList.add(child)            ▷ add to list for processing
14:                 end if
15:             end for
16:         end for
17:     end while
18:     return root                                  ▷ root is a "forest" of sentences
19: end procedure
```

Note the differences in selecting and expanding multiple sets of rules in each node. As opposed to a single rule in line 8 of Algorithm 1, here in line 6 there is an additional loop processing several rules in each node, and line 9 which selects multiple possible rules for expansion.

As with many such generators, it is not feasible to search through all conceivable derivation trees as the number of possible derivations grows exponentially and hence, cannot be explored in reasonable time. We choose to use a variation on the

beam search algorithm (Reddy, 1977). In particular, we devise a methodology for scoring intermediate derivations that suits the top-down generation.

Unlike (chart) parsers and generators which work bottom-up, and hence, have a common ground for comparison – a sub-tree covering specific subset of the input – our approach does not have such common ground to compare the partial derivations. In order to be able to score partially generated trees we used a Breadth-First approach for expanding the tree, thus, advancing all derivations so that at each step of the algorithm we compare trees of the same size and develop further only the top candidates.

Our beam-search algorithm is based on the indexes of nodes in a balanced binary tree (see Figure 5.3(a)) and the fact that the parent node index can be directly calculated from the daughter's index (Figure 5.3(b)). With this settings we defined a dynamic programming algorithm that in each step expands a list of selections of derivation rules in $1...n$ nodes (corresponding to a full-binary tree of n nodes) based on the best previous selection of derivation rules in $1...(n-1)$ nodes. As depicted in Figure 5.3(c), the sub-tree is considered full though it can have empty nodes following terminal nodes. The pseudo code for our search can be viewed in Algorithm 3.

The idea behind the algorithm is that in each iteration, sub-trees of n nodes are developed based on the previous best trees with $n-1$ nodes (line 5). Due to the binary assumption, it is trivial to retrieve the next node for expansion using the selection list (line 6). At each node the algorithm picks the best new available rules to add considering the derivation rules selected so far in ancestor nodes. The score of the new sub-tree with n nodes is trivially calculated from the previous score plus the score attached to the newly selected rule (line 8). Correspondingly, the new sub-tree consists of the selection so far plus the new selection (line 9). The score, along with the list of selected option is kept in a new intermediate list (line 10). This list is then evaluated to take the best k candidates (line 13) which are used in the next iteration of the algorithm. It is important to note that when comparing sub-trees, the *average* node score is used. In calculating the average the total score of the sub-tree is divided by the number of scoring nodes – these include non-terminals nodes (as terminal nodes have no selection in them). We use average node score to neutralize size differences between the compared trees.

Engineering note: By following this scheme, we are avoiding excessive use of memory. Instead of generating common sub-trees of leading candidate derivation, all the sub-trees are saved in the memory once. Different derivations are then distinguished by different selection paths. A common sub-tree is then merely a common prefix in the selection path of a derivation. Further more, only relevant (higher scoring) sub-trees are expanded further saving on the overall runtime of the algorithm.

The pseudo-code in Algorithm 3 addresses two challenges we had to tackle. The first is the termination of the generation (line 3). In bottom-up implementation the termination ends when a START node is encountered. In our case, going from the top down, there is no such stop condition. The optimal stop condition for our trees is having all branches of the tree ending with terminals. This is not trivial to evaluate as the terminal nodes could be at different heights in the tree and vary between sub-trees. What more, due to recursive rules it is possible to have an infinite derivation. For practical reasons, we decided to terminate generation at specific height of the subtree which can easily be derived from node index. In our work we selected an arbitrary yet reasonable height, that reflects the length of a

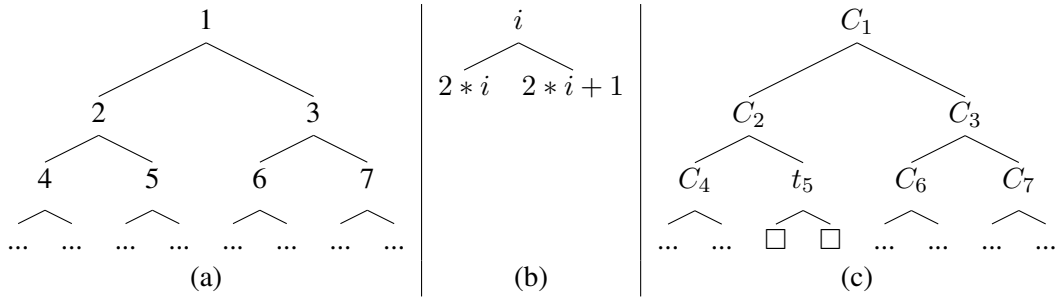


FIGURE 5.3: (a) Indexes in full binary tree and (b) The relation between parent and daughters indexes (c) Full-binary assumption is used. In case of a terminal, (t_5), all descendant nodes are marked as *empty*

moderate sentence (13-depth). In a production systems, this value can be learned from examples.

The second challenge is the scoring of selected rules during the derivation of the tree. Since our generator encapsulates all phases of generation in one algorithm, the scoring of a derivation should account for both the syntactic and semantic qualities of the resulting sentences. In addition, we want to control the emitted sentiment (and possibly other personal attributes such as emotions). Hence, the scoring function must take into consideration the syntactic emission probability as well as lexical choice, the choice of topic, and the choice of sentiment. To this end, we define three specialized grammars we discuss in the next section.

Following the generation algorithm we needed a way to select the better output sentences. While the scoring done during rule selection are meant to take into account both syntactic and semantic considerations, the complete sentences are not guaranteed to be syntactically correct or semantically coherent. Furthermore, our scoring during the generation phase is done on local non-terminal nodes and does not take into account global context.

To account for lack of control over syntactical correctness and semantic coherence, we include an additional step that sorts the output sentences based on a probabilistic language model. The underlying assumption of language modeling is that human language generation is a random process; the goal is to represent that process via a statistical model. Using a language model, we can calculate the likelihood of a words sequence being generated in the language. For our work we used n-gram models and for each sentence we calculate the overall conditional probability of the sentence. We use Microsoft's WebLM API which is part of the Microsoft Oxford Project (*Microsoft Cognitive Services*).

Selecting Rules for Generation

While the mechanics of the generation itself is relatively straight forward, when it comes to rule selection, careful decisions and tuning are required in order to achieve best results. The challenge here is to correctly select derivations so that the resulting output meets the various goals of the task – human-like, opinionated and relevant responses.

Achieving each of these goals on its own is not a simple task but the strategies for doing so are relatively clear: human-likeness or naturalness is achieved by using appropriate language constructs, hence, relates to general derivation probability; Personal or opinionated text is governed by sentiment which must be addressed

Algorithm 3 Score/Expanding Best Trees

```
1: procedure SCORETREES(root, k)
2:     currentList.add(new Payload(0,""))
3:     while stop condition not met do
4:         intermediateList = new List()
5:         for all payload in currentList do
6:             node = getNode(root,payload.selections)   ▷ (get the relevant node)
7:             for all option in node.getOptions() do     ▷ process options in node
8:                 score = payload.score + getScore(option)   ▷ get the score from
   adding this option
9:                 selection = payload.selection + option.id  ▷ the new selection is
   the
                                                              ▷ selection so far plus new
                                                              ▷ selection in current node
10:                intermediateList.add(new Payload(score, selection)) ▷ add new
   payload/candidate
11:            end for
12:        end for
13:        currentList = getTop(intermediateList, k)          ▷ select the top trees
   (selections) so far
14:    end while
15: end procedure

16: Struct Payload
17:     score                                    ▷ the score of the selection
18:     selections          ▷ the actual selection (option id) in each node so far
19: End Struct
```

throughout the generation; and relevance is closely related to topic selection or topic modeling. Selecting exactly which one of the high level scoring strategies to use in each step of the generation is the tougher research question.

The first decision to make is how to select start rules. As a rule-of-thumb we use sentiment and topic at this stage. Selecting a start rule with the non-matching sentiment to the agenda will most likely not yield the right opinion for the output. The same holds true for selecting an arbitrary lexical head. Selecting "off-topic" words would most certainly not result in a relevant text and hence, topic models should be used here as well.

Traditionally, PCFG based parsing used the occurrence probabilities in order to select the most common derivations. In generation, following a similar approach will result in more "generic" output as words or expressions commonly used would appear more and have higher probabilities. Even when expanding the derivation tree in a k-best manner and using a search strategy to find the best derivations, you more often than not have a generic result with less specific, "on-topic" verbs and nouns.

Using topic-models for selecting derivations will inherently give more relevant results – a selection based on topic-model means preferring derivations that yield words related to the topic distribution of the source document or the agenda defined for generation. The drawback for this strategy is that it may result in less natural sentences with over-usage of topic-related words and less variety.

5.2.3 Implementation Notes

In addition to the challenges of defining, extracting and then, correctly using the grammar to generate meaningful responses which meet the generation goals, we encountered other specific issues which had to be addressed in order to make the generation component operate as required and output good sentences.

First, as is the case with many CFGs, *recursive rules* are an inherent part of natural language grammar. In parsing this issue is handle through the mechanism of the decoder – unary rules are usually handled in a separate stage, and will inherently result in a lower scored derivation since they create a deeper parse with more nodes. In case of top-down generation, there is no real mechanism to prevent the recursive rules from being reselected. A recursive rule with good scores will most certainly be reselected as there is no adjacent context to prevent it. Hence, usage of recursive rules should be handle specifically, eliminating or preventing them from being re-selected repeatedly.

Next, due to the independent nature of rules selection, there is an option to either use unary or binary rule. Selecting two daughters affects the overall score compared to only one since more scoring decisions are aggregated. Depending of the scoring scheme, this will create a bias towards one type of rules or the other. In our implementation, we decided to eliminate the unary rules by connecting a child node to grandparent node, eliminating the parent's constituency and the child's dependency category. An example can be seen in Figure 5.4.

Finally, in order to be able to reconstruct some language structures we had to keep track of binarized rules. The Stanford parser outputs a binary parse tree, converting rules with more than two daughters to a chain of nodes with only two daughters. In order to be able to reconstruct these deeper derivation chains we annotated the binarized daughters with additional context from their ascendants in the same binarized chain.

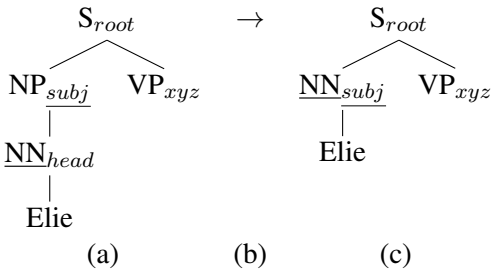


FIGURE 5.4: A chain with a unary rule is converted to eliminate the unary rule while keeping the relevant constituency of daughter – part-of-speech tag – and dependencies of the unary rule root node.

5.3 The Grammars

Automatically extracted PCFGs are now a fundamental technology in state-of-the-art probabilistic parsers. These parsers most commonly deal with phrase-structure grammars, yielding grammar in which non-terminal symbols consist of constituency categories. While this representation is sufficient for many parsing applications, when it come to language generation, requirements of the grammar changes slightly.

For both parsing and generation, the grammar may consists of phrase-structure annotation, but we see other researches in which the grammar may contain other annotations. For example, lexicalized grammar (Collins, 1997) which uses non-terminal that include both constituency category, a lexical head and a modifier; or in Cahill and Genabith (2006) where the grammar include functional categories, up-arrows that point to the f-structure associated with the mother node, and down-arrows to the local node. Other generation research such as Konstas and Lapata (2012b) define their own grammar for mapping records, fields and surface realization or Yuan, Wang, and He (2015) which uses a tailored parser that augment, or align, meaning representation with natural language expressions.

As with these works, we also need to expand go past phrase-structure grammar. In the following sections we describe our base grammar which consists of phrase-structure and sentiment annotations and then present our experimentation with adding lexical heads and dependency relations into the grammar used for generation.

5.3.1 Base Grammar

A common feature which is used throughout this work is sentiment – a main theme in our research. Each node in our grammar is annotated with sentiment class:

$$s \in [-2...2]$$

Our sentiment annotation is based on the Stanford Sentiment classification parser (Socher et al., 2013) which annotates every node in the phrase-structure parse tree with one of 5 sentiment classes. Aiming to produce opinionated/personal utterances, we think that sentiment is an obvious and relevant choice. The sentence or document level sentiment annotation which was used in the first phase give less control over the generated text. Having the sentiment annotation embedded in the grammar allows for a finer grain selection of rules when deriving a generation tree.

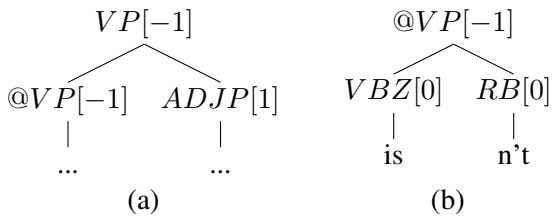


FIGURE 5.5: A subtree with changing sentiment between parent and daughters

$@VP[-1] \rightarrow VBZ[0] \;\; RB[0]$ $VBZ[0] \rightarrow is$ $RB[0] \rightarrow n't$	$@VP[-1] \rightarrow isn't$
(a)	(b)

TABLE 5.1: Regular vs fixed derivation of cases where parent has sentiment and pre-terminal daughters do not

Thus, including sentiment annotation, the non-terminals in our constituency only grammar now also includes a sentiment class, s_i, s_j and s_k for the parent and two daughters respectively:

$$C_i[\mathbf{s_i}] \rightarrow C_j[\mathbf{s_j}] \;\; C_k[\mathbf{s_k}]$$

With this grammar, rule selection can involve matching the node-level sentiment with the expected sentiment level of the overall sentence.

With node-level sentiment annotation, it is not uncommon to find changing sentiment within a rule; consider partial parse as appears in Figure 5.5 (a) and (b). Note that the sentiment of the parent node does not necessarily matches that of its daughters. Also, as in (b), the sentiment of the parent is not necessarily that of either daughters. Case (b) is also interesting as this kind of rule, even though it exists in the grammar, may result in sentiment mismatch further down the derivation.

A standard rule derivation of case (b) would include the rules in Table 5.1(a). In reality though, leaving such rules in the grammar may results in a mistake in generation as the pre-terminals, $VBZ[0]$ and $RB[0]$, do not carry any sentiment and hence, without using additional context, could emit neutral words, even though it is expected that the overall subtree would have a negative sentiment. This is fixed by handling these kind of cases specifically to produce the rule in Table 5.1(b).

5.3.2 Lexicalized Grammar

Our first modification of the basic grammar is a *lexicalized grammar* (Collins, 1997). In his research, the author suggests a generative model which includes both a constituency category and a lexical item in each non-terminal in the grammar. The motivation for that is to overcome two shortcoming of standard PCFGs: (1) Lack of Sensitivity to Lexical Information and (2) Lack of Sensitivity to Structural Preferences. While the latter issue is less relevant to our task, the former applies to generation as well: a "clean" PCFGs essentially generate lexical items as an afterthought (following generation of all the tree), conditioned only on the POS in the pre-terminals. This is a very strong independence assumption, which leads to

non-optimal decisions being made by the parser or the generator in many important cases.

In addition, for our generation task, having a lexical head in each node can help in rule selection when deriving a response tree. Instead of generating only phrase-structure derivation, which carries no lexical choice, we now introduce this knowledge into the generated sentence from the get-go, which lend itself favorably to generating a more relevant sentences.

With the addition of a lexicalized head to the nodes in our grammar, we defined the following transition rules:

$$C_i[s_i, \mathbf{l_i}] \rightarrow_1 C_j[s_j, \mathbf{l_i}] \;\; C_k[s_k, \mathbf{l_k}]$$

or

$$C_i[s_i, \mathbf{l_i}] \rightarrow_2 C_j[s_j, \mathbf{l_j}] \;\; C_k[s_k, \mathbf{l_i}]$$

where C_x is the constituency category, s_x is the sentiment and l_x is the lexical head of the parent and daughters, with $x \in \{i, k, j\}$. Note that on the right hand side there is a new lexical item, a *modifier word* in Collins (1997), which can come in the first or the second daughter. The other lexical term is the same in the parent and one of the daughters.

To obtain this grammar we used Stanford NLP library semantic head finder implementation. This is slightly modified version of the algorithm used in Collins (1997). This implementation chooses the semantic head verb rather than the verb form for cases with verbs and it makes similar themed changes to other categories (John Rappaport, 2016).

Parameters Estimation for Lexicalized Grammar

As noted in Collins (1997), there are two possible rules for a given parent: a left-rule, in which the head word is in the left daughter and the modifier on the right, $X(h) \rightarrow_1 Y_1(h) \;\; Y_2(m)$, and a right-rule, which is the opposite $X(h) \rightarrow_1 Y_1(m) \;\; Y_2(h)$. Both variants have to be accounted for in the grammar. Note that the selection of a transition rule and that of a modifier are done as two consecutive decisions. Considering the grammar above, our estimation for a left-rule emission probabilities is as follows[1].

$$P(C_j[s_j, \mathbf{l_i}]\; C_k[s_k, \mathbf{l_k}] \mid C_i[s_i, \mathbf{l_i}]) =$$
$$P(C_j[s_j, \mathbf{l_i}]\; C_k[s_k] \mid C_i[s_i, \mathbf{l_i}]) \times P(l_k | C_j[s_j, \mathbf{l_i}]\; C_k[s_k]\;,\; C_i[s_i, \mathbf{l_i}])$$

where,

$$P(C_j[s_j, \mathbf{l_i}]\; C_k[s_k] \mid C_i[s_i, \mathbf{l_i}]) = \frac{count(C_i[s_i, \mathbf{l_i}] \rightarrow_1 C_j[s_j, \mathbf{l_i}] \;\; C_k[s_k])}{count(C_j[s_j, \mathbf{l_i}])}$$

and,

$$P(l_k | C_j[s_j, \mathbf{l_i}]\; C_k[s_k]\;,\; C_i[s_i, \mathbf{l_i}]) = \frac{count(count(C_i[s_i, \mathbf{l_i}] \rightarrow_1 C_j[s_j, \mathbf{l_i}] \;\; C_k[s_k, \mathbf{l_k}])}{count(C_i[s_i, \mathbf{l_i}] \rightarrow_1 C_j[s_j, \mathbf{l_i}] \;\; C_k[s_k])}$$

[1] And vise versa for a right rule

5.3.3 Lexicalized Relational Realizational Grammar

Including lexical heads makes our grammar a bit more useful when creating responses. Instead of a "generic" phrase-structure derivation, with lexicalized grammar we are able to take into account the lexical choice throughout the generation process and not only at the pre-terminals; this allow us to fine-tune selection and get more relevant responses. Still, we would like our grammar to be more functional: phrase-structure brings in *form* – how sentences are built, but we would like to have some insight into grammatical functions of the phrase. Subject, object, tenses and grammatical gender could help us refine the response – and these concept cannot be capture by constituency and lexical head alone.

Dependency grammar (Tesnière, 1959) captures that required information. This grammar consist of a graph of binary dependencies between syntactically or semantically related words in a sentence. The drawback of dependency grammar with respect to generation is that it is un-ordered, putting no constraint on word order. As such, it does not lend itself easily for generating.

Relational-Realizational (RR) grammar (Tsarfaty and Sima'an, 2008), combines phrase-structure and dependency annotation in one representation giving both form and function in a single parse tree.

Following the RR approach, our new grammar now looks as follows:

$$C_i[s_i, \mathbf{dep_i}, l_i] \rightarrow_1 C_j[s_j, \mathbf{dep_j}, l_i] \;\; C_k[s_k, \mathbf{dep_k}, l_k]^2$$

Essentially, we have added to each node a functional category, $dep_x | x \in \{i, j, k\}$, which determines its functional role with relation to its parent. With our RR grammar, we augment the phrase-structure and sentiment of the parent with a functional component which affect the selection of daughters. Our process of selecting the next rules also follows the original RR grammar definitions: in the first stage, *Projection*, we generate a set of grammatical relations between the children to their parent node; in the second stage, *Configuration*, these relations are ordered; and in the last stage, *Realization*, the daughters' constituency, sentiment and lexical head are selected taking required functional role into account.

To obtains a lexicalized RR trees we have followed the algorithm described in (Tsarfaty, Nivre, and Andersson, 2011). Given both a constituency parse and a dependency graph of the sentence, we follow a deterministic process of converting the dependency graph into a dependency tree and then merge it with the lexicalized phrase-structure tree. Merging is done based on matching spans over words within the sentence. The algorithms for converting the dependency graph and merging it with constituency parse is available in the aforementioned citation. Examples of phrase-structure, dependency and corresponding RR parses of the same sentence are presented in Figure 5.6.

Parameters Estimation for RR

Given the lexicalized RR trees with phrase-structure, dependency, sentiment and lexical head annotation, we can induce the lexicalized RR grammar. Following the definitions in Tsarfaty and Sima'an (2008), we induce three distinct rule-sets and occurrence probabilities, corresponding to the three stages of Projection, Configuration and Realization.

[2] or the corresponding rule $C_i[s_i, \mathbf{dep_i}, l_i] \rightarrow C_j[s_j, \mathbf{dep_j}, l_j] \;\; C_k[s_k, \mathbf{dep_k}, l_i]$, note l_i is in the second daughter

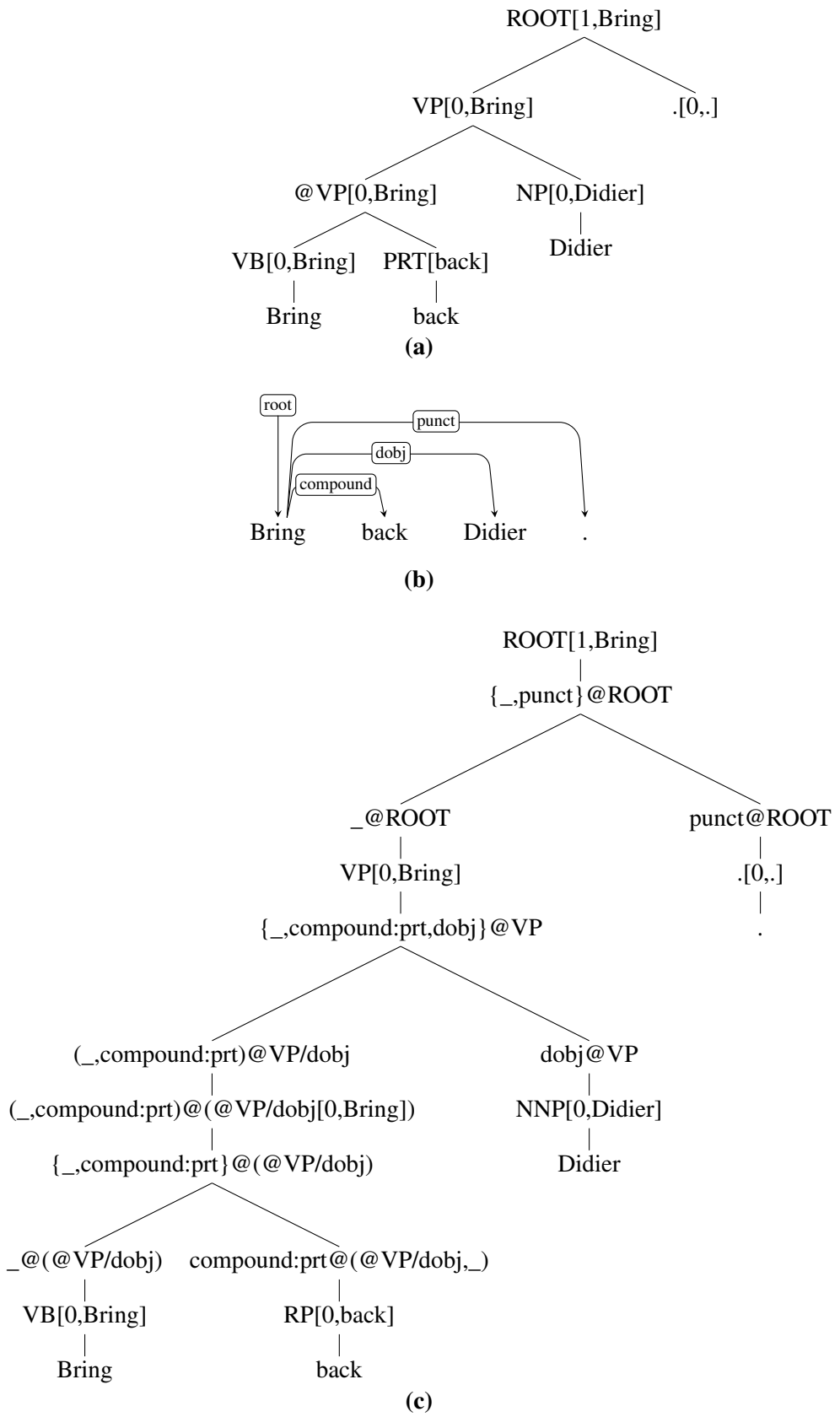


FIGURE 5.6: Annotated Phrase-Structure Tree (a) and Dependency Graph (b) parses of the same sentence, and the corresponding lexicalized RR tree (c). Note that nodes with $[s_i, l_i]$ corresponds to the nodes in the phrase-structure tree and the other nodes are the projection and configuration of the RR.

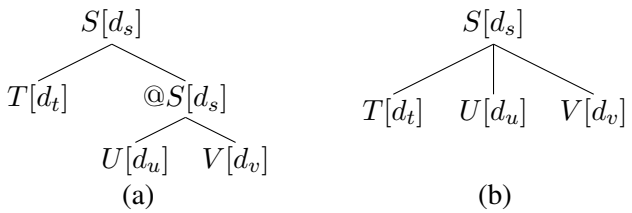


FIGURE 5.7: A binarized nodes derivation (a) and the corresponding non-binarized one (b).

For the projection rules we consider the constituency and sentiment of the parent and looking for the dependencies of the daughters:

$$P(\{dep_j\}_{j\in deps(i)}|[C_i, s_i]) = \frac{count(\{dep_j\}_{j\in deps(i)})}{count(C_i[s_i])}$$

Note that here we consider not only the dependency annotation of the immediate daughters but the dependencies of the descendants of i which are under the same original constituency category before it was binarized. From Figure 5.7(b) one can see that for the root node, $S[d_s]$ the original dependencies includes $[d_t, d_v, d_v]$ even though in the binarized tree (a) there are only two daughters.

The configuration estimates slightly differs from Tsarfaty and Sima'an (2008) in that where the original grammar allows for any number of daughters, we only allow two. Given a set of dependencies to realize, in the multi-daughter implementation each dependency can be realized directly while we have to keep only two daughters to accommodate our search algorithm. This means we have to realize two *sets*, one containing one dependency and the other the rest of the projected dependencies. Thus, the estimate considers the phrase-structure category and sentiment as above as well as the projection, and emit two sets:

$$P(< dep_j >_{j\in deps(i)/k}, < dep_k > |C_i[s_i], \{dep_j\}_{j\in deps(i)}) = \frac{count(< dep_j >_{j\in deps(i)/k}, < dep_k >)}{count(C_i[s_i], \{dep_j\}_{j\in deps(i)})}$$ [3]

Our realization estimates also varies from Tsarfaty and Sima'an (2008). The original article only deals with realizing constituency, but we also have sentiment and lexical head to realize. Hence, our realization is broken into two steps, each entailing its own estimation and each realizing a set of 1 or more dependencies, namely, $< dep_j >_{j\in deps(i)/k}$ and $< dep_k >$.

The first realization step selects the daughter's constituency and sentiment $C_j[s_j]$ (or $C_k[s_k]$) given the parent constituency and sentiment, $C_i[s_i]$, and the set of dependencies to realized as determined in previous step. Note that we have two sets to realize:

$$P(C_j[s_j]|C_i[s_i], < dep_j >_{j\in deps(i)/k}) = \frac{count(C_j[s_j])}{count(C_i[s_i], < dep_j >_{j\in deps(i)/k})}$$

and

$$P(C_k[s_k]|C_i[s_i], < dep_k >) = \frac{count(C_k[s_k])}{count(C_i[s_i], < dep_k >)}$$

[3] the the sets could also be reversed, $< dep_k >, < deps_j >_{j\in deps(i)/k}$

The second realization step realizes the lexical head of the daughter, l_j (or l_k), considering the constituency and sentiment that were realized and the parent's constituency, sentiment and lexical head.

$$P(l_j|C_i[s_i,l_k],< dep_j >_{j\in deps(i)/k},C_j[s_j]) = \frac{count(l_j)}{count(C_i[s_i,l_i],< dep_j >_{j\in deps(i)/k},C_j[s_j])}$$

and

$$P(l_k|C_i[s_i,l_k],< dep_k >,C_j[s_j]) = \frac{count(l_j)}{count(C_i[s_i,l_i],< dep_k >,C_j[s_j])}$$

5.4 Evaluation

In the previous section we introduced three grammars which are designed for opinionated generation. Given these grammars we want to evaluate which grammar performs best. Our evaluation follows the architecture described in Section 5.2 – we collected a dataset, induced the three grammars and then used them to generate many responses. The generation was done with each grammar and with several sentiment levels. In part of the experiements we addressed different source documents. The resources used in the evaluation are described in Section 5.4.1.

Our evaluation followed two tracks: an automated assessment of the quality of each grammars along a few criteria which is described in Section 5.4.2, and a Turing-like test similar to the one used for evaluating the template-based responses in Chapter 4 and is described in Section 5.4.3

5.4.1 Materials

The first resource required for evaluation is a dataset that can be used for inducing grammar and for training topic model. Since no relevant dataset exist, we created a new corpus of news articles and corresponding user comments from a news site. To collect the data for our corpus we developed scripts that retrieves user comments from the NY-Times®web site through their open Community API. All-in-all we collected 151,256 random comments that were published between 2009-06 and 2015-03. We then collected the corresponding news articles from the NY-Times site itself using a simple crawler, and came up with 2,344 articles for the corresponding period and comments. In addition, to supplement the data-set we collected additional 3,099 random articles for general use (e.g. topic modeling).

Next, for our experiment, we used a subset of the corpus to induce the grammars. We selected to focus on sports news, which gave us around 3,600 news articles and 13,100 user comments that included a total of 55,700 sentences. The sentences were parsed and processed following the procedures described earlier to give us the three grammars defined in the previous section.

Finally, we simulated several generation scenarios, in order to test various aspects of the grammars and the corresponding generated responses. In each execution we set the system to generate sentences with different grammars or scoring schemes. The results of each simulation are 5,000 responses for each variant of the system, consisting of 1,000 sentences for each sentiment level in $s \in -2,-1,0,1,2$.

Grammar	Sentiment	Sentence
PCFG	-2	(and badly should doesn't..
	-1	doesn't of the yankees..
	0	who is the the game,.
	1	is the the united states..
	2	is the best players..
LEX	-2	is a rhyme ... mahi mahi, and, I not quote Bunny.
	-1	Dumpster unpire are the villans.
	0	Derogatory big names symbols wider
	1	New england has been playful, and infrequent human.
	2	That's a huge award – having get fined!
RR	-2	he is very awkward, and to any ridiculous reason.
	-1	the malfeasance underscores the the widespread belief.
	0	the programs serve the purposes.
	1	McIlroy is a courageous competitor.
	2	The urgent service's a grand idea.

TABLE 5.2: Responses generated by the system with the different grammars and sentiment levels.

5.4.2 Automatic Measures

We conducted two experiments using automatic measures. In **Experiment 1** we compared the three grammars, trying to evaluate them along three criteria: (i) Compactness, (ii) Fluency, and (iii) Sentiment Agreement. In **Experiment 2** we tested one of the grammars for relevance. The experiments are described in the following sections.

Experiment 1: Comparing Grammars

In our first experiment we compared the grammars specified in Section 5.3, in order to try and identify the one that yields the best sentences. We evaluated 3 aspects of the responses: *Compactness*, *Fluency* and *Sentiment Agreement*. Example sentences from this experiments are presented in Table 5.2.

- **Compactness** is a measure of how efficient the grammar is in capturing the language. A more compact grammar will realize the concepts with less complex and shallower tress. To evaluate the Compactness of the grammar we observed out of the 5000 generated sentences, which trees where complete – that is – have all the children in the tree being terminal symbols (words) [4]. As we see in Table 5.3, the relational realizational grammar out performed the other two grammars, yielding compete sentences in more than 95% of the responses. This means the RR Grammar capture the language in more straight forward manner, giving less complex derivation trees, which also lends itself for better performance and more control over the generation.

- **Fluency** measure is an indication of how grammatical or natural the sentences are. Based on joint probability distribution, this measure gives an indication of how common are word-sequences within the sentence. For

[4] We set the generator for trees of maximum depth of 13. This give a potential of up to 4096 words. Inreality, the realization was of much shorter sentences and depending on the grammar, some incomplete realizations.

Grammar	Avg. LM Score		Avg. LM Score per word		Complete Sentences (%)	Sentiment Agreement (%)	Sentiment Polarity (%)	Avg. Length (words)
	Mean	CI	Mean	CI				
PCFG	-79.677	±0.054	-8.937	±0.007	20.080	13.280	41.760	9.537
LEX	-73.702	±0.016	-6.534	±0.002	67.260	**44.620**	**63.880**	12.275
RR	**-51.747**	±0.011	**-5.559**	±0.001	**95.740**	43.840	60.960	9.628
HUMAN	-50.062	±0.000	-5.443	±0.000	N/A	N/A	N/A	10.26

TABLE 5.3: Mean and 95% confidence interval of language model scores, and measures of compactness and sentiment agreement. The last row, *HUMAN* compare the corresponding measurements for human responses collected online.

this evaluation we used Microsoft Web ML API which return the aggregated minus-log probabilities of all 3-grams in the sentence. Using the reported joint probability we also calculated a normalized, per-word, sentence score. As can be seen in Table 5.3 the RR grammar give better results with average per word joint probability of -5.559, compare with -6.534 for Lexicalized grammar and -8.937 for PCFG. Also, for the complete sentences only, the language model score of the RR grammar is much better showing that this grammar yield more common language construct than the other two for any size sentences.

- **Sentiment Agreement** is a measure of whether or not the perceived sentiment of the response matches the input sentiment level parameter used in the generation. We want to get responses which match the input sentiment level. Sentiment Polarity assessment is a more relaxed measure that only takes into account the sign (negative, neutral or positive) of the sentiment. We see in Table 5.3 that the most accurate grammar is the lexicalized grammar. It out-scores RR grammar by a few percent in both sentiment and sentiment polarity. This shows that lexicalized grammar was more sensitive to the input sentiment. It is important to note here that the sentiment was put into play only in the initial selection of a start rules at the beginning of the generation.

Experiment 2: Testing Relevance

Following the opinionated NLG theme, we tested for relevance of the responses. More specifically, taking the RR grammar we wanted to test whether the use of topic model will yield responses which are more closely related to the source document. For the purpose of this test we define *Topic Agreement* to be a measure that, given a specific trained topic model, determines how close the topic distribution of the source document and the response are. We used L2-metric to calculate the distance between the two inferred topic distribution vectors.

For the purpose of this test we used responses from two generators, both using the RR grammar. The first generator, same as in the previous experiment, selects the start rules based on occurrence probabilities. The second generator, *RRTM*, uses topic model to give a score based on the head words of the daughters of the start rule:

Grammar	Sentiment	Sentence
RR	-2	they deserve it, but I is fear.
	-1	the saga is correct.
	0	the indirect penalty?
	1	the job is correct.
	2	a salaries excels.
RRTM	-2	Unfortunately, they remind that to participate in baseball.
	-1	the franchise would he made?
	0	Probably the LONG time .
	1	In a good addition, he is a good baseball player.
	2	the baseball game sublime.

TABLE 5.4: Responses generated by the system using emission probabilities and topic models for the start rule selection.

Generator	Mean	CI
RR	0.473	$\pm$ 0.003
RRTM	0.424	$\pm$ 0.003
HUMAN	0.429	$\pm$ 0.000

TABLE 5.5: Mean and 95% confidence interval for generators with and without topic models usage (RRTM and RR respectively). The last row, *HUMAN* compare the corresponding measurements for human responses collected online.

$$score(START \rightarrow \beta) = \sum_{t=1}^{N} \sum_{i=1}^{2} tm\_weight(t) * word\_weight(t, l_i)$$

where $tm\_weight(t)$ is the weight of topic t in the topic distribution of the source document, and $word\_weight(t, l_i)$ is the weight of the lexical head word l_i within the word distribution weights of topic t in the given topic model. The inner sum traverse all the daughters in β, namely for binary rules, 1 and 2 .

The results of the two generators and their average distance from the topic distribution of the source document are presented in Table 5.5. As can be seen in the table, the generator using topic model for selecting start rules (RRTM) gets topic distribution that is closer to the source's topic distribution. The last row, *HUMAN* is the average distance for all the sentences used to induce the grammars. Since these sentences come from whole paragraphs is makes sense that some of the sentences are connectives and other auxiliary language and hence the RRTM model "out-performed" it.

5.4.3 Surveys

Similar to the template-based generation evaluation, we performed a human-likeness evaluation of the grammars by collecting data via an online surveys on Amazon Mechanical Turk (www.mturk.com). In the survey, participants were asked to judge whether generated sentences were written by human or computer, akin to (a simplified version of) the Turing test (Turing, 1950).

Grammar	Mean	CI
PCFG	2.4561	± 0.004
LEX	4.1681	± 0.004
RR	3.7278	± 0.004

TABLE 5.6: Mean and 95% confidence interval for the human-likeness rating of each grammar.

The participant were pre-screened through a simple qualification test that asked "At which age did you start to learn English, either from your parents/caregivers or in school?". Only participants who started to learn English at or before the age of four were allowed to participate, ensuring a good level of English proficiency. We also required that the participant were from the USA.

Each survey comprised of 50 randomly ordered trials. In each trial the survey-taker was shown a random sentence. Over the 50 trials the participant was exposed to three or four sentences from each grammar and sentiment level combination. The task was to categorize each response on a 7-point scale with labels 'Certainly human/computer', 'Probably human/computer', 'Maybe human/computer' and 'Unsure'. The average human-likeness score is listed in Table 5.6. In this survey it is clear that the sentences generated by using the lexicalized grammar were perceived as most human like. This results is in contrast of the automatic evaluation we performed though this results is not surprising as noted by others before (see Section 2.4 for relevant discussion).

In addition to mean and CI we also run an ordinal mixed-effects regression, which is the theoretically correct way to analyze rating data. This is especially true considering the survey results are far from Gaussian distribution. In Table 5.7 we report the regression analysis results.

The results of the regression analysis shows the following trends:

1. Relative to Lex, PCFG is much less human-like (b=-2.89; p<.0001) and RR is a bit less human-like (b=-0.57; p<.0001).

2. The effects (1) are modulated by sentiment: more positive sentiment makes both PCFG and RR more human-like relative to Lex (respectively: b=0.18; p=.06, and b=0.62, p<.0001).

3. The effects of (1) are also modulated by sentence length in #words: longer sentences make PCFG less human-like (b=-1.30; p<.0001) but RR human-likeness is not affected by sentence length.

4. There is no main effect of sentiment (SENT) on human-likeness.

5. Longer sentences are considered less human-like (b=-0.29; p<.0001) and this is particularly the case for PCFG (see (3))

6. The position estimate have little effect on the score (b=0.21, p<0.001). This means there is no learning effect throughout the survey. This is also modulated by sentence length. The positive effect of this estimate on the scores is most likely related to raters recalibrating their judgment as the survey progress. We believe this is more of an indicative of a psychological phenomena than something that is relevant to modeling sentence generation.

The interpretation of b-coefficients for (1) is the average difference in rating compared to Lex grammar. For (2),(3),(5) b equals the amount of change in rating

Factor	b	Std. Error	z-value	$P(>\|z\|)$
G-PCFG	-2.89835	0.18925	-15.315	< 2e-16
G-RR	-0.57038	0.09933	-5.742	9.34e-09
SENT	-0.01116	0.06155	-0.181	0.85606
NWORD	-0.29009	0.05025	-5.773	7.77e-09
POS	0.21482	0.03596	5.974	2.32e-09
G-PCFG × SENT	0.18174	0.09519	1.909	0.05623
G-RR × SENT	0.61786	0.08811	7.012	2.34e-12
G-PCFG × NWORD	-1.30530	0.11700	-11.156	< 2e-16
G-RR × NWORD	0.04585	0.10145	0.452	0.65130
NWORD × POS	0.10391	0.03697	2.811	0.00494

TABLE 5.7: Regression analysis results of the human-likeness survey.

for one standard deviation change in the relevant predictor if all other predictors have their average value.

5.5 Conclusions

In this chapter we introduced a data-driven approach to opinionated text generation. We defined a framework for grammar-based generation of natural language that is both human-like and opinionated. The framework includes grammar extraction and an algorithm for using the grammar along with sentiment, topic and PCFG-like mechanism to generate opinionated sentences in social communication context.

Following the collection of a new data-set for our task, we defined a basic sentiment-augmented grammar and examined how this can be expanded for creating more relevant and informative grammars. We explored generation with three types of grammars (i) a baseline, unlexicalized, PCFG, (ii) a lexicalized grammar which includes a lexical head and modifier, and (iii) a relational-realizational grammar which also factor in dependency annotation to make the overall generation more coherent. In addition we took into account sentiment, topic modeling and general phrase structure probabilities to obtain better, more coherent and fluent, sentences for the opinionated generation task.

We evaluated the grammars with both automatic assessment and human rating. In the automated assessment we show that RR grammar was more compact and fluent based on language model while the lexicalized resulted in better sentiment-agreement. Also, we show that using topic models for rule selection yields better relevance for the grammar. In the human evaluation we learn that the lexicalized grammar was perceived as more human-like and that longer sentences are less human-like while input sentiment levels did not affect human-likeness rating.

Chapter 6

Discussion and Future Work

In this research we explored *opinionated* natural language generation, a relatively new field in NLG. We presented a model centered around a *user model* and *analysis* of online *documents* for *topic* and *sentiment*. Following this model, we implemented two generation systems: a simple, *template-based* and mostly hand-coded proof-of-concept system, and a more robust, *data-driven* architecture which uses induced grammars for generation of responses..

The template-based system served as a feasibility test which proved successful, as our results show close measure of identification as real responses in human-likeness and relevance measures. That phase of the research shows us that world-knowledge improves the human-likeness of responses, and that the limited ability of the template-based approach to produce a large diversity of responses limits its feasibility for use in an open domain.

We followed the first phase with a data-driven grammar-based implementation that jointly models opinions and structure in the grammar, and uses it in a generative process that combines micro-planning and surface realization. The system uses grammar to construct a derivation tree and then realizes it. In this second stage we collected a new, novel dataset for the task, defined three types of grammars and evaluated them, using both automated assessment and through human rating. We showed that the Relational-Realizational grammar scores better in automatic fluency assessment and that the lexicalized grammar was conceived as more human-like by human raters.

The combined knowledge from these two contributions points us toward a more fine-tuned usage of the data-driven, grammar-based generator. We saw that using sentiment and topics modeling can generate the required overall opinionated responses but requires a more careful, and deeper realization within the generation process, whether in a refined grammar or through more subtle scoring schemes.

What follows is a list of points which we believe are instrumental for improving further the generation of subjective, opinionated responses.

- **World Knowledge**: Our result concerning the human-likeness of g_{kb} (Section 4.3.3) clearly demonstrates that semantic knowledge must be brought in to support better, and more human-like, response generation. In general, our efforts in this work are more toward micro-planning. Large-scale knowledge graphs such as Freebase[1], for extracting such world knowledge, support many semantic tasks (Jacobs, 1985), and can be used for providing richer context for automatically generating human-like responses.

- **Functional Grammar**: The current grammar uses a concrete lexicalization as part of its annotation. In order to get a more general and flexible generator, these annotations can be generalized by replacing the concrete words

[1] http://www.freebase.com

with abstract concepts. Such grammars would use phrase-structure, dependency and sentiment annotation along with semantic placeholders, yielding an abstract representation that can open the door for a more refined lexical usage with selection of referring expressions, verbs and so on. A natural place to start exploring such grammars could be using POS tags or Named Entity Recognition (NER, Grishman and Sundheim (1996)) to replace the lexical annotation. Another approach could be integration of Frame-semantic parses (Das et al., 2014b) into the grammar inducing steps.

- **Candidates Search Procedure**: While our approach for data-driven generation of opinionated responses is novel, the search algorithm we employed for getting the best candidates is relatively simple. The grammar was used to directly score the derived candidates based on a relatively small set of parameters reflecting the different generative stories. In addition, in the schema we developed, language models could only be applied at the end of the generation. This framework can be expanded to use a more involved scoring and ranking, based on advanced features that will be used during generation. Another approach is using modified chart generation (Haruno, Den, and Matsumoto, 1996) instead of the top-down approach we have used. This will allow integration of a language-model within the generation, evaluating the surface realization incrementally, at each iteration of the search process.

- **Macro Planning**: In this research we mainly dealt with micro-planning, which was done implicitly in during derivation rule selection. The generator selected the lexical items to be used, and hence, determined the topics while generating the response tree. In future work it would be interesting to also explore the macro planning stage constructing paragraphs as opposed to sentences. As we have shown in the first phase (Section 4.3.3), the use of a knowledge-base to expand the response increases its overall human-likeness. A macro-planning phase guided by data driven knowledge-bases and ontologies could yield more interesting and relevant results.

- **Personal and Personalized Generation**: In this work, we have only used sentiment for making a personal and opinionated response. While this is a major part of personal communication – having sentiment, or agenda toward topics – we believe that exploring emotions, cynicism and other such language features can push the perceived human-likeness, and hence, the usability of such a technology. Furthermore, while we presented an approach to generate personal utterances, the technique can be further refined to be personalized to specific responders. Employing the same pipeline but using a specific corpus (for example, one that is extracted from a specific user interaction in a social network) could result in a generator that mimics a specific person.

- **Interactive Framework**: Our current development may be conceived as static in the sense that it generates responses to a specific static document. For real interpersonal communication an opinionated agent should be able to interact in a meaningful way in an ongoing dialog. Adapting our algorithm to have a temporal state representation including both semantic and mood would be a challenging yet interesting task. This can be expanded with some real-time performance constraint and adapted to Text-to-Speech applications, for examples, interaction with voice-controlled toys and other human-computer interfaces.

- **New Evaluation Methodologies**: A recent article (Zarrieß, Loth, and Schlangen, 2015) shows that reading times can predict the quality of generated text beyond the capabilities of human rating. For more accurate results we can adapt such methodology for evaluation of our generated text without relying on subjective opinions of raters. Another interesting approach is evaluation of the generated text in online context. After posting the text online, we can measure interactions with it, such as responses and shares. Such real-world evaluation could indicate that generated responses are indeed believable and engaging, and may better simulate a Turing-like test in which machine-generated responses cannot be distinguished from human responses.

- **A Theoretical/Social Investigation**: From a theoretical viewpoint, the system will clearly benefit from rigorous analysis of human interaction in online media. Responses to user-generated content on the Internet share some linguistic characteristics in structure, length and manner of expression. Studying these features theoretically and then examining them empirically using a Turing-like evaluation as presented here can take us a big step in the direction of better generation, and also better understanding of the processes underlying human response generation.

  This latter understanding may be complemented with insights into the causes, motivations and intricacies of human interaction in such environments, as studied by sociologists and psychologists. Such insights could be used to refine our user model and its intersection with content from the document. In particular, our preliminary interaction with colleagues from communication studies suggests that the present endeavor nicely complements that of "persuasive computing" (Fogg, 1998; Fogg, 2002), and we hope that this collaboration will lead to valuable synergies.

Chapter 7

Conclusion

In this thesis we presented a novel task, *opinionated* natural language generation, and different ways to approach it. Thus, the contribution of this thesis is manifold.

First, we introduced the task of *opinionated NLG*, described its settings, gave it interdisciplinary theoretical grounding and defined the high level components that are required to solve it. At its core, we defined an online user and its interaction in online social context. We put forth an architecture and defined its main components including an *analysis* step for retrieving context from an online document and a *generation* step which intersects the document context and the user model in order to generate opinionated texts.

Next, we provided two general architectures for generation of opinionated responses. Initially we developed a working template-based proof-of-concept generation system, performed a thorough evaluation of it through a new evaluation methodology, and draw interesting conclusions as to what make online responses human-like and relevant. The results of the evaluation shows that world-knowledge improves the human-likeness of responses, and that the limited ability of the template-based approach to produce a large diversity of responses limits its usefulness in open domains.

Following that, we developed a new data-driven grammar-based approach which was designed to overcome the major shortcoming of the template-based approach. In this phase, we developed and evaluated three types of grammars – a vanilla PCFG, a lexicalized PCFG, and a relational realizational grammar – which are all also augmented with sentiment annotation in all non-terminal nodes. We defined a general search strategy through the generated response candidates. We have shown the relative strength of each grammar and the overall usefulness of sentiment and topic selection in such settings. As part our contribution we also released a new decorated data set for inducing grammars.

Our results provide new insights concerning key differences between human-generated and computer-generated responses, in the hope that this inspires further research and more sophisticated models for *Opinionated NLG research*.

Bibliography

Amabile, Teresa M. (1981). *Brilliant but Cruel: Perceptions of Negative Evaluators*. English. Washington, DC: ERIC Clearinghouse, p. 28. URL: http://www.eric.ed.gov/contentdelivery/servlet/ERICServlet?accno=ED211573.

Aman, Saima and Stan Szpakowicz (2007). "Identifying Expressions of Emotion in Text". In: *TSD*. Ed. by Václav Matousek and Pavel Mautner. Vol. 4629. Lecture Notes in Computer Science. Springer, pp. 196–205. ISBN: 978-3-540-74627-0.

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani (2010). "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining". In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA). ISBN: 2-9517408-6-7.

Bates, Douglas M. (2005). "Fitting linear mixed models in R". In: *R News* 5, pp. 27–30.

Becker, Tilman (2002). "Practical, template-based natural language generation with TAG". In: *Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*.

Belz, Anja and Ehud Reiter (2006). "Comparing Automatic and Human Evaluation of NLG Systems". In: *In Proc. EACL'06*, pp. 313–320.

Blei, David M. (2012). "Probabilistic Topic Models". In: *Commun. ACM* 55.4, pp. 77–84. ISSN: 0001-0782. DOI: 10.1145/2133806.2133826. URL: http://doi.acm.org/10.1145/2133806.2133826.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2003). "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3, pp. 993–1022. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=944919.944937.

Booth, T. L. and R. A. Thompson (1973). "Applying Probability Measures to Abstract Languages". In: *IEEE Transactions on Computers* C-22.5, pp. 442–450. ISSN: 0018-9340. DOI: 10.1109/T-C.1973.223746.

Busemann, Stephan and Helmut Horacek (1998). "A Flexible Shallow Approach to Text Generation". In: *CoRR* cs.CL/9812018. URL: http://dblp.uni-trier.de/db/journals/corr/corr9812.html#cs-CL-9812018.

Cagan, Tomer, Stefan L. Frank, and Reut Tsarfaty (2014). "Generating Subjective Responses to Opinionated Articles in Social Media: An Agenda-Driven Architecture and a Turing-Like Test". In: *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*. Baltimore, Maryland: Association for Computational Linguistics, pp. 58–67. URL: http://www.aclweb.org/anthology/W/W14/W14-2708.

Cahill, Aoife and Josef van Genabith (2006). "Robust PCFG-based Generation Using Automatically Acquired LFG Approximations". In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. ACL-44. Sydney, Australia: Association for Computational Linguistics, pp. 1033–1040. DOI: 10.3115/1220175.1220305. URL: http://dx.doi.org/10.3115/1220175.1220305.

Callison-Burch, Chris and Mark Dredze (2010). "Creating Speech and Language Data with Amazon's Mechanical Turk". In: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. CSLDAMT '10. Los Angeles, California: Association for Computational Linguistics, pp. 1–12. URL: http://dl.acm.org/citation.cfm?id=1866696.1866697.

Chaudhri, Vinay K. et al. (2006). "A Case Study in Engineering a Knowledge Base for an Intelligent Personal Assistant". In: *Proceedings of the 5th International Conference on Semantic Desktop and Social Semantic Collaboration - Volume 202*. SemDesk'06. Athens, GA: CEUR-WS.org, pp. 25–32. URL: http://dl.acm.org/citation.cfm?id=2889986.2889989.

Chomsky, N. (1957). *Syntactic structures*. Janua linguarum: Minor. Mouton. URL: https://books.google.co.il/books?id=CyxZAAAAMAAJ.

Collins, Michael (1997). "Three Generative, Lexicalised Models for Statistical Parsing". In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. ACL '98. Madrid, Spain: Association for Computational Linguistics, pp. 16–23. DOI: 10.3115/976909.979620. URL: http://dx.doi.org/10.3115/976909.979620.

Dale, Robert and Ehud Reiter (1995). "Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions". In: *Cognitive Science* 19.2, pp. 233–263. ISSN: 1551-6709. DOI: 10.1207/s15516709cog1902_3. URL: http://dx.doi.org/10.1207/s15516709cog1902_3.

Danescu-Niculescu-Mizil, Cristian et al. (2009). "How Opinions Are Received by Online Communities: A Case Study on Amazon.Com Helpfulness Votes". In: *Proceedings of the 18th International Conference on World Wide Web*. WWW '09. Madrid, Spain: ACM, pp. 141–150. ISBN: 978-1-60558-487-4. DOI: 10.1145/1526709.1526729. URL: http://doi.acm.org/10.1145/1526709.1526729.

Das, Dipanjan et al. (2014a). "Frame-Semantic Parsing". In: *Computational Linguistics* 40:1, pp. 9–56.

— (2014b). "Frame-Semantic Parsing". In: *Computational Linguistics* 40:1, pp. 9–56.

Davidov, Dmitry, Oren Tsur, and Ari Rappoport (2010). "Enhanced Sentiment Learning Using Twitter Hashtags and Smileys". In: *Proceedings of the 23rd International Conference on Computational Linguistics*. Beijing, China: Association for Computational Linguistics, pp. 241–249. URL: http://dl.acm.org/citation.cfm?id=1944566.1944594.

Dempster, Martin, Norman Alm, and Ehud Reiter (2010). "Automatic Generation of Conversational Utterances and Narrative for Augmentative and Alternative Communication: A Prototype System". In: *Proceedings of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies*. SLPAT '10. Los Angeles, California: Association for Computational Linguistics, pp. 10–18. URL: http://dl.acm.org/citation.cfm?id=1867750.1867752.

DeVault, David, David Traum, and Ron Artstein (2008). "Practical Grammar-based NLG from Examples". In: *Proceedings of the Fifth International Natural Language Generation Conference*. INLG '08. Salt Fork, Ohio: Association for Computational Linguistics, pp. 77–85. URL: http://dl.acm.org/citation.cfm?id=1708322.1708338.

Ekman, Paul (1999). *Basic Emotions*. New York, NY: John Wiley & Sons Ltd., pp. 45–60.

Elhadad, Michael and Jacques Robin (1998). *SURGE: a Comprehensive Plug-in Syntactic Realization Component for Text Generation*. Tech. rep. ACL.

Feng, Donghui et al. (2006). "An Intelligent Discussion-Bot for Answering Student Queries in Threaded Discussions". In: *Proceedings of Intelligent User Interface (IUI-2006)*, pp. 171–177.

Fogg, B. J. (1998). "Persuasive Computers: Perspectives and Research Directions". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '98. Los Angeles, California, USA: ACM Press/Addison-Wesley Publishing Co., pp. 225–232. ISBN: 0-201-30987-4. DOI: 10.1145/274644.274677. URL: http://dx.doi.org/10.1145/274644.274677.

— (2002). "Persuasive Technology: Using Computers to Change What We Think and Do". In: *Ubiquity*. ISSN: 1530-2180. DOI: 10.1145/764008.763957. URL: http://doi.acm.org/10.1145/764008.763957.

Foster, Mary Ellen (2008). "Automated Metrics That Agree with Human Judgements on Generated Output for an Embodied Conversational Agent". In: *Proceedings of the Fifth International Natural Language Generation Conference*. INLG '08. Salt Fork, Ohio: Association for Computational Linguistics, pp. 95–103. URL: http://dl.acm.org/citation.cfm?id=1708322.1708341.

Frawley, William J., Gregory Piatetsky-Shapiro, and Christopher J. Matheus (1992). "Knowledge Discovery in Databases: An Overview". In: *AI Mag.* 13.3, pp. 57–70. ISSN: 0738-4602. URL: http://dl.acm.org/citation.cfm?id=140629.140633.

Gatt, Albert and Ehud Reiter (2009). "SimpleNLG: A Realisation Engine for Practical Applications". In: *Proceedings of the 12th European Workshop on Natural Language Generation*. ENLG '09. Athens, Greece: Association for Computational Linguistics, pp. 90–93. URL: http://dl.acm.org/citation.cfm?id=1610195.1610208.

Grice, H. P. (1967). "Logic and conversation". In: *Studies in the ways of words*. Ed. by H. P. Grice. Harvard University Press, pp. 22–40.

Grishman, Ralph and Beth Sundheim (1996). "Message Understanding Conference-6: A Brief History". In: *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*. COLING '96. Copenhagen, Denmark: Association for Computational Linguistics, pp. 466–471. DOI: 10.3115/992628.992709. URL: http://dx.doi.org/10.3115/992628.992709.

Haenlein, Michael and Andreas M. Kaplan (2009). "Flagship Brand Stores within Virtual Worlds: The Impact of Virtual Store Exposure on Real-Life Attitude toward the Brand and Purchase Intent". In: *Recherche et Applications en Marketing (English Edition)* 24.3, pp. 57–79. DOI: 10.1177/205157070902400303. eprint: http://rme.sagepub.com/content/24/3/57.full.pdf+html. URL: http://rme.sagepub.com/content/24/3/57.abstract.

Haruno, Masahiko, Yasuharu Den, and Yuji Matsumoto (1996). "Trends in Natural Language Generation An Artificial Intelligence Perspective: Fourth European Workshop, EWNLG '93 Pisa, Italy, April 28–30, 1993 Selected Papers". In: ed. by Giovanni Adorni and Michael Zock. Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. A chart-based semantic head driven generation algorithm,

pp. 300–313. ISBN: 978-3-540-49457-7. DOI: 10.1007/3-540-60800-1_36. URL: http://dx.doi.org/10.1007/3-540-60800-1_36.

Hasegawa, Takayuki et al. (2013). "Predicting and Eliciting Addressee's Emotion in Online Dialogue". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 964–972. URL: http://www.aclweb.org/anthology/P13-1095.

Hays, David G. (1964). "Dependency Theory: A Formalism and Some Observations". In: *Language* 40.4, pp. 511–525. ISSN: 00978507, 15350665. URL: http://www.jstor.org/stable/411934.

Hofmann, Thomas (1999). "Probabilistic Latent Semantic Indexing". In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '99. Berkeley, California, USA: ACM, pp. 50–57. ISBN: 1-58113-096-1. DOI: 10.1145/312624.312649. URL: http://doi.acm.org/10.1145/312624.312649.

Howard, Philip N. et al. (2011). "Opening Closed Regimes: What Was the Role of Social Media During the Arab Spring?" In: *Project on Information Technology and Political Islam*. URL: http://pitpi.org/index.php/2011/09/11/opening-closed-regimes-what-was-the-role-of-social-media-during-the-arab-spring/.

Jacobs, Paul S (1985). *A Knowledge-Based Approach to Language Production*. Tech. rep. Berkeley, CA, USA: University of California at Berkeley.

John Rappaport Marie-Catherine de Marneffe, Anna Rafferty (2016). *Stanford CoreNlp Documentation*. URL: http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/trees/SemanticHeadFinder.html.

Kathuria, Pulkit (2012). *Sentiment Classification using WSD, Maximum Entropy and Naive Bayes Classifiers. https://github.com/kevincobain2000/sentiment_classifier*. Visited March 2014. URL: http://www.jaist.ac.jp/~s1010205/sentiment_classifier.

Kay, Martin (1996). "Chart Generation". In: *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*. ACL '96. Santa Cruz, California: Association for Computational Linguistics, pp. 200–204. DOI: 10.3115/981863.981890. URL: http://dx.doi.org/10.3115/981863.981890.

Klein, Dan and Christopher D. Manning (2003). "Accurate Unlexicalized Parsing". In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*. ACL '03. Sapporo, Japan: Association for Computational Linguistics, pp. 423–430. DOI: 10.3115/1075096.1075150. URL: http://dx.doi.org/10.3115/1075096.1075150.

Konstas, Ioannis and Mirella Lapata (2012a). "Concept-to-text Generation via Discriminative Reranking". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*. ACL '12. Jeju Island, Korea: Association for Computational Linguistics, pp. 369–378. URL: http://dl.acm.org/citation.cfm?id=2390524.2390576.

— (2012b). "Unsupervised Concept-to-text Generation with Hypergraphs". In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, pp. 752–761. URL: http://www.aclweb.org/anthology/N12-1093.

— (2013). "A Global Model for Concept-to-Text Generation". In: *Journal of Artificial Intelligence Research* 48, pp. 305–346.

Krcadinac, U. et al. (2013). "Synesketch: An Open Source Library for Sentence-Based Emotion Recognition". In: *Affective Computing, IEEE Transactions on* 4.3, pp. 312–325. ISSN: 1949-3045. DOI: 10.1109/T-AFFC.2013.18.

Lamer, Wiebke (2012). "Twitter and Tyrants: New Media and its Effects on Sovereignty in the Middle East". In: *Arab Media and Society* (16). ISSN: 1687-7721. URL: http://www.arabmediasociety.com/?article=798.

Langheinrich, Marc and Günter Karjoth (2011). "Social Networking and the Risk to Companies and Institutions". In: *Information Security Technical Report. Special Issue: Identity Reconstruction and Theft*, pp. 51–56. DOI: http://dx.doi.org/10.1016/j.istr.2010.09.001. URL: http://www.elsevier.com/wps/find/journaldescription.cws_home/31185/description.

Langner, Brian (2010). "Data-driven Natural Language Generation: Making Machines Talk Like Humans Using Natural Corpora". AAI3528166. PhD thesis. Pittsburgh, PA, USA. ISBN: 978-1-267-58209-6.

Lavie, Alon and Abhaya Agarwal (2007). "Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments". In: *Proceedings of the Second Workshop on Statistical Machine Translation*. StatMT '07. Prague, Czech Republic: Association for Computational Linguistics, pp. 228–231. URL: http://dl.acm.org/citation.cfm?id=1626355.1626389.

Lester, James (1994). "Generating Natural Language Explanations from Large-Scale Knowledge Bases". PhD thesis. Austin, TX: Department of Computer Science, University of Texas at Austin. URL: http://www.cs.utexas.edu/users/ai-lab/?lester:phd94.

Lester, James C. and Bruce Porter (1997). "Developing and Empirically Evaluating Robust Explanation Generators: The KNIGHT Experiments". In: *Computational Linguistics Journal* 23.1, pp. 65–101. URL: http://www.cs.utexas.edu/users/ai-lab/?lester:clj97.

Li, Haifang et al. (2007). "Research on textual emotion recognition incorporating personality factor". In: *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pp. 2222–2227. DOI: 10.1109/ROBIO.2007.4522515.

Mani, Inderjeet and Mark T. Maybury (2001). "Automatic Summarization". In: *Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter, Companion Volume to the Proceedings of the Conference: Proceedings of the Student Research Workshop and Tutorial Abstracts, July 9-11, 2001, Toulouse, France*. P. 5.

Mann, William C. (1983). "An overview of the PENMAN text generation system". In: *Proceedings of the National Conference on Artificial Intelligence*. Also appears as USC/Information Sciences Institute, RR-83-114. AAAI, pp. 261–265.

Marcu, Daniel (2000). *The Theory and Practice of Discourse Parsing and Summarization*. Cambridge, MA, USA: MIT Press. ISBN: 0262133725.

Microsoft Cognitive Services. URL: https://www.microsoft.com/cognitive-services/en-us/web-language-model-api.

Mishne, Gilad (2006). "Multiple ranking strategies for opinion retrieval in blogs". In: *Proceedings of the 15th Text Retrieval Conference*.

Mori, Kyoshi, Adam Jatowt, and Mitsuru Ishizuka (2003). "Enhancing Conversational Flexibility in Multimodal Interactions with Embodied Lifelike Agent".

In: *Proceedings of the 8th International Conference on Intelligent User Interfaces*. IUI '03. Miami, Florida, USA: ACM, pp. 270–272. ISBN: 1-58113-586-6. DOI: 10.1145/604045.604096. URL: http://doi.acm.org/10.1145/604045.604096.

Nadeau, David and Satoshi Sekine (2007). "A survey of named entity recognition and classification". In: *Linguisticae Investigationes* 30.1. Publisher: John Benjamins Publishing Company, pp. 3–26. URL: http://www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002.

Narayan, Karthik Sankaran, Charles Lee Isbell Jr., and David L. Roberts (2011). "DEXTOR: Reduced Effort Authoring for Template-Based Natural Language Generation." In: *Proceedings of the Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*. Ed. by Vadim Bulitko and Mark O. Riedl. The AAAI Press. URL: http://dblp.uni-trier.de/db/conf/aiide/aiide2011.html#NarayanIR11.

Niekrasz, John et al. (2005). "Ontology-based discourse understanding for a persistent meeting assistant". In: *In Proceedings of the 2005 AAAI*. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.148.4440.

Pang, Bo and Lillian Lee (2005). "Seeing Stars: Exploiting Class Relationships For Sentiment Categorization With Respect To Rating Scales". In: *Proceedings of ACL*, pp. 115–124.

— (2008). "Opinion Mining and Sentiment Analysis". In: *Found. Trends Inf. Retr.* 2.1-2. Interested in 4.1.2 Subjectivity Detection and Opinion Identification", pp. 1–135. ISSN: 1554-0669. DOI: 10.1561/1500000011. URL: http://dx.doi.org/10.1561/1500000011.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan (2002). "Thumbs Up? Sentiment Classification Using Machine Learning Techniques". In: *Proceedings of EMNLP*, pp. 79–86.

Papadimitriou, Christos H. et al. (1998). "Latent Semantic Indexing: A Probabilistic Analysis". In: *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. PODS '98. Seattle, Washington, USA: ACM, pp. 159–168. ISBN: 0-89791-996-3. DOI: 10.1145/275487.275505. URL: http://doi.acm.org/10.1145/275487.275505.

Papineni, Kishore et al. (2002). "BLEU: a Method for Automatic Evaluation of Machine Translation". In: pp. 311–318.

Paris, Cecile et al. (2007). "Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation". In: ed. by Robert Dale and Michael White. Chap. Desiderata for Evaluation of Natural Language Generation, 9–16.

Paul, Michael J. and Roxana Girju (2010). "A Two-Dimensional Topic-Aspect Model for Discovering Multi-Faceted Topics." In: *AAAI*. Ed. by Maria Fox and David Poole. AAAI Press. URL: http://dblp.uni-trier.de/db/conf/aaai/aaai2010.html#PaulG10.

Pavlopoulos, John and Ion Androutsopoulos (2014). "Multi-Granular Aspect Aggregation in Aspect-Based Sentiment Analysis". In: *Proceedings of the14th Conference of the European Chapter of the Association for Computational Linguistics*. EACL '14. Gothenburg, Sweden: European Association for Computational Linguistics, pp. 78–87.

Piskorski, Jakub and Roman Yangarber (2013). "Information Extraction: Past, Present and Future". English. In: *Multi-source, Multilingual Information Extraction and Summarization*. Ed. by Thierry Poibeau et al. Theory and Applications

of Natural Language Processing. Springer Berlin Heidelberg, pp. 23–49. ISBN: 978-3-642-28568-4. DOI: 10.1007/978-3-642-28569-1_2. URL: http://dx.doi.org/10.1007/978-3-642-28569-1_2.

Qualman, Erik (2012). *Socialnomics: How social media transforms the way we live and do business*. 2nd. Hoboken, NJ, USA: John Wiley & Sons. ISBN: 978-1118232651. URL: http://books.google.com/books?hl=en\&\#38;lr=\&\#38;id=fH075AmvTVUC\&\#38;oi=fnd\&\#38;pg=PT13\&\#38;dq=social+media\&\#38;ots=ZlrCFUSaB0\&\#38;sig=ctJ3KaIl_PbE1i1pH-vBPaFk16E.

Reddy, D. Raj (1977). *Speech understanding systems: summary of results of the five-year research effort at Carnegie-Mellon University.* Tech. rep. Carnegie-Mellon University.

Rehurek, Radim and Petr Sojka (2010). "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.

Reilly, W. Scott et al. (1996). *Believable Social and Emotional Agents*.

Reiter, Ehud (2011). "Task-Based Evaluation of NLG Systems: Control vs Real-World Context". In: *Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop*. Edinburgh, Scotland: Association for Computational Linguistics, pp. 28–32. URL: http://www.aclweb.org/anthology/W11-2704.

Reiter, Ehud and Anja Belz (2009). "An Investigation into the Validity of Some Metrics for Automatically Evaluating Natural Language Generation Systems". In: *Comput. Linguist.* 35.4, pp. 529–558. ISSN: 0891-2017. DOI: 10.1162/coli.2009.35.4.35405. URL: http://dx.doi.org/10.1162/coli.2009.35.4.35405.

Reiter, Ehud and Robert Dale (1997). "Building Applied Natural Language Generation Systems". In: *Nat. Lang. Eng.* 3.1, pp. 57–87. ISSN: 1351-3249. DOI: 10.1017/S1351324997001502. URL: http://dx.doi.org/10.1017/S1351324997001502.

— (2000). *Building Natural Language Generation Systems*. New York, NY, USA: Cambridge University Press. ISBN: 0-521-62036-8.

Reiter, Ehud, Somayajulu Sripada, and Roma Robertson (2003). "Acquiring Correct Knowledge for Natural Language Generation." In: *J. Artif. Intell. Res. (JAIR)* 18, pp. 491–516. URL: http://dblp.uni-trier.de/db/journals/jair/jair18.html#ReiterSR03.

Reiter, Ehud et al. (2009). "Using NLG to Help Language-impaired Users Tell Stories and Participate in Social Dialogues". In: *Proceedings of the 12th European Workshop on Natural Language Generation*. ENLG '09. Athens, Greece: Association for Computational Linguistics, pp. 1–8. URL: http://dl.acm.org/citation.cfm?id=1610195.1610196.

Ritter, Alan, Colin Cherry, and William B. Dolan (2011). "Data-driven Response Generation in Social Media". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, pp. 583–593. ISBN: 978-1-937284-11-4. URL: http://dl.acm.org/citation.cfm?id=2145432.2145500.

Ritter, Alan et al. (2011). "Named Entity Recognition in Tweets: An Experimental Study". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, pp. 1524–1534. ISBN: 978-1-937284-11-4. URL: http://dl.acm.org/citation.cfm?id=2145432.2145595.

Robin, Jacques (1994). "Automatic Generation and Revision of Natural Language Report Summaries Providing Historical Background". In: *In Proceedings of the 11th Brazilian Symposium on Artificial Intelligence*.

Rus, Vasile et al. (2011). "Question Generation Shared Task and Evaluation Challenge: Status Report". In: *Proceedings of the 13th European Workshop on Natural Language Generation*. ENLG '11. Nancy, France: Association for Computational Linguistics, pp. 318–320. URL: http://dl.acm.org/citation.cfm?id=2187681.2187740.

Shieber, Stuart M. et al. (1989). "A Semantic-head-driven Generation Algorithm for Unification-based Formalisms". In: *Proceedings of the 27th Annual Meeting on Association for Computational Linguistics*. ACL '89. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 7–17. DOI: 10.3115/981623.981625. URL: http://dx.doi.org/10.3115/981623.981625.

Socher, Richard et al. (2013). "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, WA: Association for Computational Linguistics, pp. 1631–1642.

Stone, Matthew et al. (2001). "Microplanning with Communicative Intentions: The SPUD System". In: *CoRR* cs.CL/0104022. URL: http://arxiv.org/abs/cs.CL/0104022.

Strong, Christina R. et al. (2007). "Emotionally Driven Natural Language Generation for Personality Rich Characters in Interactive Games". In: *AIIDE*. Ed. by Jonathan Schaeffer and Michael Mateas. Stanford, California, USA: The AAAI Press, pp. 98–100. ISBN: 978-1-57735-325-6. URL: http://www.aaai.org/Papers/AIIDE/2007/AIIDE07-021.pdf.

Tan, Chenhao et al. (2011). "User-Level Sentiment Analysis Incorporating Social Networks". In: *Proceedings of KDD*, pp. 1397–1405.

Tesnière, L (1959). *Elements de syntaxe structurale*. Ed. by Editions Klincksieck. Editions Klincksieck.

Tesnière, Lucien (1959). *Eléments de Syntaxe Structurale*. Paris: Klincksieck.

Theune, M. et al. (2001). "From Data to Speech: A General Approach". In: *Nat. Lang. Eng.* 7.1, pp. 47–86. ISSN: 1351-3249. URL: http://dl.acm.org/citation.cfm?id=973927.973930.

Titov, Ivan and Ryan McDonald (2008). "Modeling Online Reviews with Multi-grain Topic Models". In: *Proceedings of the 17th International Conference on World Wide Web*. WWW '08. Beijing, China: ACM, pp. 111–120. ISBN: 978-1-60558-085-2. DOI: 10.1145/1367497.1367513. URL: http://doi.acm.org/10.1145/1367497.1367513.

Tsarfaty, Reut, Joakim Nivre, and Evelina Andersson (2011). "Evaluating Dependency Parsing: Robust and Heuristics-Free Cross-Annotation Evaluation". In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, pp. 385–396. URL: http://www.aclweb.org/anthology/D11-1036.

Tsarfaty, Reut and Khalil Sima'an (2008). "Relational-realizational Parsing". In: *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*. COLING '08. Manchester, United Kingdom: Association for Computational Linguistics, pp. 889–896. ISBN: 978-1-905593-44-6. URL: http://dl.acm.org/citation.cfm?id=1599081.1599193.

Turing, Alan M. (1950). "Computing Machinery and Intelligence". In: *Mind* LIX, pp. 433–460.

Van Deemter, Kees, Emiel Krahmer, and Mariët Theune (2005). "Real Versus Template-Based Natural Language Generation: A False Opposition?" In: *Comput. Linguist.* 31.1, pp. 15–24. ISSN: 0891-2017. DOI: 10.1162/0891201053630291. URL: http://dx.doi.org/10.1162/0891201053630291.

Viswanath, Bimal et al. (2009). "On the Evolution of User Interaction in Facebook". In: *Proceedings of the 2nd ACM Workshop on Online Social Networks*. WOSN '09. Barcelona, Spain: ACM, pp. 37–42. ISBN: 978-1-60558-445-4. DOI: 10.1145/1592665.1592675. URL: http://doi.acm.org/10.1145/1592665.1592675.

Wilson, Theresa et al. (2005). "OpinionFinder: A System for Subjectivity Analysis". In: *Proceedings of HLT/EMNLP on Interactive Demonstrations*. HLT-Demo '05. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 34–35. DOI: 10.3115/1225733.1225751. URL: http://dx.doi.org/10.3115/1225733.1225751.

Wimalasuriya, Daya C. and Dejing Dou (2010). "Ontology-based Information Extraction: An Introduction and a Survey of Current Approaches". In: *J. Inf. Sci.* 36.3, pp. 306–323. ISSN: 0165-5515. DOI: 10.1177/0165551509360123. URL: http://dx.doi.org/10.1177/0165551509360123.

Wu, Chung-Hsien, Ze-Jing Chuang, and Yu-Chung Lin (2006). "Emotion Recognition from Text Using Semantic Labels and Separable Mixture Models". In: *Robotics and Biomimetics* 5.2, pp. 165–183. ISSN: 1530-0226. DOI: 10.1145/1165255.1165259. URL: http://doi.acm.org/10.1145/1165255.1165259.

Young, R. Michael (1999). "Using Grice's Maxim of Quantity to Select the Content of Plan Descriptions". In: *Artificial Intelligence* 115, pp. 215–256.

Yuan, Caixia, Xiaojie Wang, and Qianhui He (2015). "Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)". In: Brighton, UK: Association for Computational Linguistics. Chap. Response Generation in Dialogue Using a Tailored PCFG Parser, pp. 81–85. URL: http://aclweb.org/anthology/W15-4713.

Zarrieß, Sina and Jonas Kuhn (2013). "Combining Referring Expression Generation and Surface Realization: A Corpus-Based Investigation of Architectures". In: *ACL*.

Zarrieß, Sina, Sebastian Loth, and David Schlangen (2015). "Reading Times Predict the Quality of Generated Text Above and Beyond Human Ratings". In: *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*. Brighton, UK: Association for Computational Linguistics, pp. 38–47. URL: http://www.aclweb.org/anthology/W15-4705.

Generating Subjective Responses to Opinionated Articles in Social Media: An Agenda-Driven Architecture and a Turing-Like Test

Tomer Cagan
School of Computer Science
The Interdisciplinary Center
Herzeliya, Israel
cagan.tomer@idc.ac.il

Stefan L. Frank
Centre for Language Studies
Radboud University
Nijmegen, The Netherlands
s.frank@let.ru.nl

Reut Tsarfaty
Mathematics and Computer Science
Weizmann Institute of Science
Rehovot, Israel
tsarfaty@weizmann.ac.il

Abstract

Natural language traffic in social media (blogs, microblogs, talkbacks) enjoys vast monitoring and *analysis* efforts. However, the question whether computer systems can *generate* such content in order to effectively interact with humans has been only sparsely attended to. This paper presents an architecture for generating subjective responses to opinionated articles based on users' agenda, documents' topics, sentiments and a knowledge graph. We present an empirical evaluation method for quantifying the human-likeness and relevance of the generated responses. We show that responses generated using world knowledge in the input are regarded as more human-like than those that rely on topic, sentiment and agenda only, whereas the use of world knowledge does not affect perceived relevance.

1 Introduction

Digital media, user-generated content and social networks enable effective human interaction; so much so that much of our day-to-day interaction is conducted online (Viswanath et al., 2009). Interaction in social media fundamentally changes the way businesses and consumers behave (Qualman, 2012), can be instrumental to the success of individuals and businesses (Haenlein and Kaplan, 2009), and even affects the stability of political regimes (Howard et al., 2011; Lamer, 2012). These facts force organizations (businesses, governments, and non-profit organizations) to be constantly involved in the monitoring of, and the interaction with, human agents in digital environments (Langheinrich and Karjoth, 2011).

Automatic analysis of user-generated online content benefits from extensive research and commercial opportunities. In natural language processing, there is ample research on the analysis of subjectivity and sentiment of content in social media. The development of tools for sentiment analysis (Davidov et al., 2010), mood aggregation (Agichtein et al., 2008), opinion mining (Mishne, 2006), and many more, now enjoys wide interest and exposure, as is also evident by the many workshops and dedicated tracks at ACL venues.[1] Methods are also developed for the analysis of political texts (O'Connor et al., 2010; O'Connor et al., 2013) and for text-driven forecasting based on these data (Yano et al., 2009). A related strand of research uses computational methods to find out what kind of published utterances are influential, and how they affect linguistic communities (Danescu-Niculescu-Mizil et al., 2009). Such work complements, and contributes to, studies from sociology and sociolinguistics that aim to delineate the process of generating meaningful responses (e.g., Amabile (1981)).

In contrast to these analysis efforts, the topic of *generating* responses to content in social media is only sparsely explored. Commercially, there is movement towards online response automation (Owyang, 2012; Mah, 2012).[2] Research on user interfaces is trying to move away from script-based interaction towards the development of chat bots that attempt natural human-like interaction (Mori et al., 2003; Feng et al., 2006). However, these chat bots are typically designed to provide an automated one-size-fits-all type of interaction.

A study by Ritter et al. (2011) addresses the generation of responses to natural language tweets in a data-driven setup. It applies a machine-translation approach to response generation, where moods and sentiments already ex-

[1] E.g., the ACL series LASM http://tinyurl.com/ludyrkz; WASSA http://tinyurl.com/kjjdhax.

[2] There is a general debate on the efficiency of automated tools (Nall, 2013) and whether such tools are desirable in social media (McConnell (2012); responses to Owyang (2012)).

pressed in the past are replicated or reused. A recent study by Hasegawa et al. (2013) modifies Ritter's approach to produce responses that elicit an emotion from the addressee. Yet, these responses do not target particular topics and are not driven by a user agenda.

The present paper addresses the problem of generating novel, subjective, responses to online opinionated articles. We formally define the document-to-response mapping problem and suggest an end-to-end system to solve it. Our system integrates a range of NLP and NLG technologies (including topic models, sentiment analysis, and the integration of a knowledge graph) to design a flexible generation mechanism that allows us to vary the information in the input to the generation procedure. We then use a Turing-inspired test to study the different factors that contribute to the perceived human-likeness and relevance of the generated responses, and show how the perception of responses depends on external knowledge and the expressed sentiment.

The remainder of this paper is organized as follows. The next section presents our proposal: Section 2.1 describes our approach, Section 2.2 formalizes the proposal, and Section 2.3 presents our end-to-end architecture. This is followed by our evaluation method and empirical results in Section 3. We discuss related and future work in Section 4, and in Section 5 we conclude.

2 The Proposal: Generating Subjective Responses

2.1 Our Approach

Natural language is, above all, a communicative device that we employ to achieve certain goals. In social media, the driving force behind generating responses is a responder's disposition towards some topic. This topic could be a political campaign or a candidate, a product, or some abstract idea, which the responder has a motive to promote. Let us call this goal our user's *agenda*.

User response generation, like any other natural language utterance generation, is triggered by a certain event that is related to the communicative goal. In a social media setting, this event is often a new online *document*. The document and the agenda thus form the input to our generation system. Each document and each agenda contain (possibly many) topics, each of which is associated with a (positive or negative) sentiment. Document sentiments are attributed to the author, whereas agenda sentiments are attributed to the user (henceforth: the responder).

For each non-empty intersection of the topics in the document and in the agenda, our response-generation system aims to generate utterances that are fluent, human-like, and effectively engage readers. The generation is based on three assumptions, roughly reflecting the Gricean maxims of cooperative interaction (Grice, 1967). Online user responses should then be:

- *Economic* (Maxim of Quantity): Responses are brief and concise;
- *Relevant* (Maxim of Relation): Responses directly address the documents' content.
- *Opinionated* (Maxim of Quality): Responses express responders beliefs, sentiments, or dispositions towards the topic(s).

2.2 The Formal Model

Let D be a set of documents and let A be a set of user agendas as we define shortly. Let S be a set of English sentences over a finite vocabulary $S = \Sigma^*$. Our system implements a function that maps each $\langle document, agenda \rangle$ pair to a natural language response sentence $s \in S$.

$$f_{\text{response}} : D \times A \rightarrow S$$

Response generation takes place in two phases, roughly corresponding to macro and micro planning in Reiter and Dale (1997):

- Macro Planning (below, the *analysis* phase): What are we going to talk about?
- Micro Planning (below, the *generation* phase): How are we going to say it?

The analysis function $p : D \rightarrow C$ maps a document to a subjective representation of its content.[3] The generation function $g : C \times A \rightarrow S$ intersects the content elements in the document and in the user agenda, and generates a response based on the content of the intersection. All in all, our system implements a composition of the analysis and the generation functions:

$$f_{\text{response}}(d, a) = g(p(d), a) = s$$

[3] A content element may conceivably encompass a topic, its sentiment, its objectivity, its evidentiality, its perceived truthfulness, and so on. In this paper we focus on topic and sentiment, and leave the rest for future research.

Each content element $c \in C$ or an agenda item $a \in A$ is composed of a topic t associated with a sentiment value $sentiment_t \in [-n..n]$ that signifies the (negative or positive) disposition of the document's author (if $c \in C$) or the user's agenda (if $a \in A$) towards the topic. We assume here that a topic is simply a bag of words from our vocabulary Σ. Thus, we have the following:

$$A, C \subseteq \mathcal{P}(\Sigma) \times [-n..n]$$

Our generation component accepts the result of the intersection as input and relies on a template-based grammar and a set of functions for generating referring expressions in order to construct the output. To make the responses *economic*, we limit the content of a response to one statement about the document or its author, followed by a statement on the relevant topic. To make the response *relevant*, the templates that generate the response make use of topics in the intersection of the document and the agenda. To make the response *opinionated*, the sentiment of the response depends on the (mis)match between the sentiment values for the topic in the document and in the agenda. Concretely, the response is positive if the sentiments for the topic in the document and agenda are the same (both positive or both negative) and it is negative otherwise.

We suggest two variants of the generation function g. The basic variant implements the baseline function defined above:

$$g_{\text{base}}(c, a) = s$$

$$c \in C, a \in A, s \in \Sigma^*$$

For the other variant we define a knowledge base (KB) as a directed graph in which words $w \in \Sigma$ from the topic models correspond to nodes in the graph, and relations $r \in R$ between the words are predicates that hold in the real world. Our second generation function now becomes:

$$g_{\text{kb}}(c, a, \mathit{KB}) = s$$

$$\mathit{KB} \subseteq \{(w_i, r, w_j) | w_i, w_j \in \Sigma, r \in R\}$$

with $c \in C, a \in A, s \in \Sigma^*$ as defined in g_{base} above.

2.3 The Architecture

The system architecture from a bird's eye view is presented in Figure 1. In a nutshell, a document enters the analysis phase, where topic inference and sentiment scoring take place, resulting in ⟨topic, sentiment⟩-pairs. During the subsequent generation phase, these are intersected with the ⟨topic, sentiment⟩-pairs in the user agenda. This intersection, possibly augmented with a knowledge graph, forms the input for a template-based generation component.

Analysis phase For the task of inferring the topics of the document we use topic modeling: a probabilistic generative modeling technique that allows for the discovery of abstract topics over a large body of documents (Papadimitriou et al., 1998; Hofmann, 1999; Blei et al., 2003). Specifically, we use topic modeling based on *Latent Dirichlet Allocation* (LDA) (Blei et al., 2003; Blei, 2012). Given a new document and a trained model, the inference method provides a weighted mix of topics for that document, where each topic is represented as a vector containing keywords associated with probabilities. For training the topic model and inferring the topics in new documents we use *Gensim* (Rehurek and Sojka, 2010), a fast and easy-to-use implementation of LDA.

Next, we wish to infer the sentiment that is expressed in the text with relation to the topic(s) identified in the document. We use the semantic/lexical method as implemented in Kathuria (2012). We rely on a WSD sentiment classifier that uses the SentiWordNet (Baccianella et al., 2010) database and calculates the positivity and negativity scores of a document based on the positivity and negativity of individual words. The result of the sentiment analysis is a pair of values, indicating the positive and negative sentiments of the document-based scores for individual words. We use the larger of these two values as the sentiment value for the whole document.[4]

Generation phase Our generation function first intersects the set of topics in the document and the set of topics in the agenda in order to discover relevant topics to which the system would generate responses. A response may in principle integrate content from a range of topics in the topic model distribution, but, for the sake of generating concise responses, in the current implementation we focus on the single most prevalent, topic. We pick the highest scoring word of the highest scoring topic, and intersect it with topics in the agenda. The system generates a response based on the identified

[4]Clearly, this is a simplifying assumption. We discuss this assumption further in Section 4.

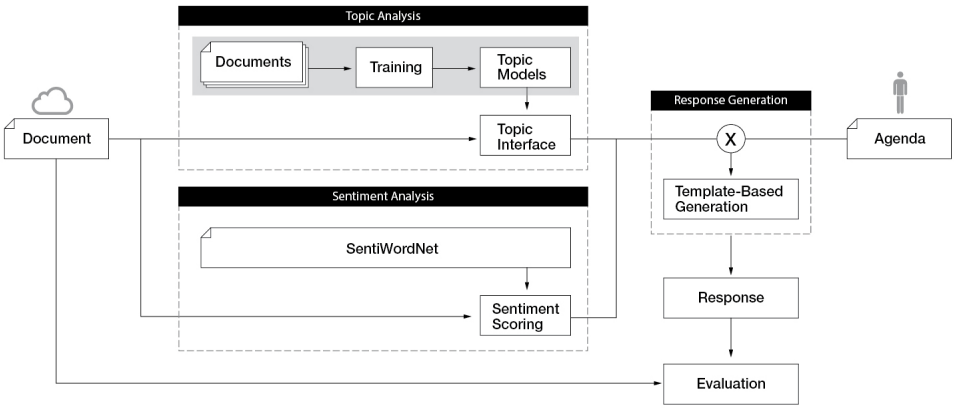


Figure 1: The system architecture from a bird's eye view. Components on gray background are executed offline.

topic, the sentiment for the topic in the document, and the sentiment for that topic in the user agenda.

The generation component relies on a template-based approach similar to Reiter and Dale (1997) and Van Deemter et al. (2005). Templates are essentially subtrees with leaves that are placeholders for other templates or for functions generating referring expressions (Theune et al., 2001). These functions receive (relevant parts of) the input and emit the sequence of fine-grained part-of-speech (POS) tags that realizes the relevant referring expression. The POS tags in the resulting sequences are ultimately place holders for words from a lexicon Σ. In order to generate a variety of expression forms — nouns, adjectives and verbs — these items are selected randomly from a fine-grained lexicon we defined. The sentiment (positive or negative) is expressed in a similar fashion via templates and randomly selected lexical entries for the POS slots, after calculating the overall sentiment for the intersection as stated above. Our generation implementation is based on SimpleNLG (Gatt and Reiter, 2009) which is a surface realizer API that allows us to create the desired templates and functions, and aggregates content into coherent sentences. The templates and functions that we defined are depicted in Figure 2.

In addition, we handcrafted a simple knowledge graph (termed here KB) containing the words in a set of pre-defined user agendas. Table 1 shows a snippet of the constructed knowledge graph. The knowledge graph can be used to expand the response in the following fashion: The topic of the response is a node in the KB. We randomly select one of its outgoing edges for creating a related

Source	Relation	Target
Apple	CompetesWith	Samsung
Apple	CompetesWith	Google
Apple	Creates	iOS

Table 1: A knowledge graph snippet.

statement that has the target node of this relation as its subject. The related sentence generation uses the same template-based mechanism as before. In principle, this process may be repeated any number of times and express larger parts of the KB. Here we only add one single knowledge-base relation per response, to keep the responses concise.

3 Evaluation

We set out to evaluate how computer-generated responses compare to human responses in their perceived *human-likeness* and *relevance*. More in particular, we compare different system variants in order to investigate what makes responses seem more human-like or relevant.

3.1 Materials

Our empirical evaluation is restricted to topics related to mobile telephones, specifically Apple's iPhone and devices based on the Android operating system. We collected 300 articles from leading technology sites in the domain to train the topic models on, settling on 10 topics models. Next, we generated a set of user agendas referring to the same 10 topics. Each agenda is represented by a single keyword from a topic model distribution and a sentiment value $sentiment_t \in \{-8, -4, 0, 4, 8\}$. Finally, we selected 10 new articles from similar sites and generated a pool of

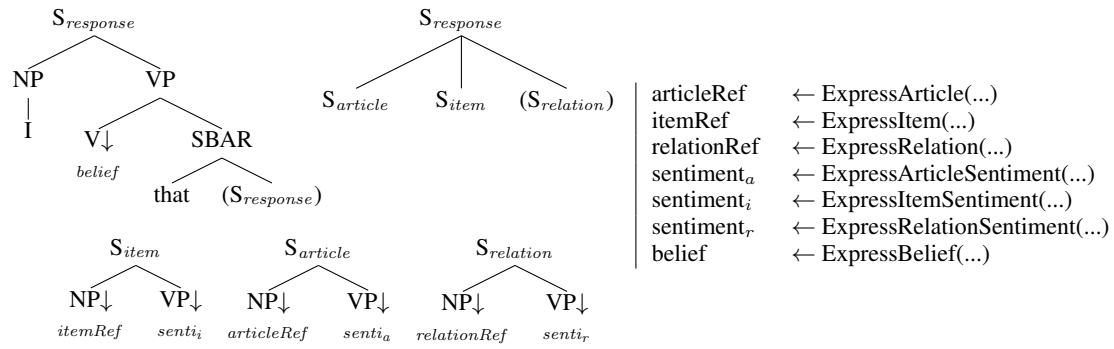


Figure 2: Template-based response generation. The templates are on the left. The Express* functions on the right uses regular expressions over the arguments and vocabulary items from a closed lexicon.

1000 responses for each, comprising 100 unique responses for each combination of $sentiment_t$ and system variant (i.e., with or without a knowledge base). Table 2 presents an example response for each such combination. In addition, we randomly collected 5 to 10 real, short or medium-length, online human responses for each article.

3.2 Surveys

We collected evaluation data via two online surveys on Amazon Mechanical Turk (`www.mturk.com`). In Survey 1, participants judged whether responses to articles were written by human or computer, akin to (a simplified version of) the Turing test (Turing, 1950). In Survey 2, responses were rated on their relevance to the article, in effect testing whether they abide by the Gricean Maxim of Relation. This is comparable to the study by Ritter et al. (2011) where people judged which of two responses was 'best'.

Each survey comprises 10 randomly ordered trials, corresponding to the 10 selected articles. First, the participant was presented with a snippet from the article. When clicking a button, the text was removed and its presentation duration recorded. Next, a multiple-choice question asked about the snippet's topic. Data on a trial was discarded from analysis if the participant answered incorrectly or if the snippet was presented for less than 10 msec per character; we took these to be cases where the snippet was not properly read. Next, the participant was shown a randomly ordered list of responses to the article.

In Survey 1, four responses were presented for each article: three randomly selected from the pool of human responses to that article and one generated by our system. The task was to categorize each response on a 7-point scale with labels 'Certainly human/computer', 'Probably human/computer', 'Maybe human/computer' and 'Unsure'. In Survey 2, five responses were presented: three human responses and two computer-generated. The task was to rate the responses' relevance on a 7-point scale labeled 'Completely (not) relevant', 'Mostly (not) relevant', 'Somewhat (not) relevant', and 'Unsure'. As a control condition, one of the human responses and one of the computer responses were actually taken from another article than the one just presented. In both surveys, the computer-generated responses presented to each participant were balanced across sentiment levels and generation functions (g_{base} and g_{kb}).

After completing the 10 trials, participants provided basic demographic information, including native language. Data from non-native English speakers was discarded. Surveys 1 and 2 were completed by 62 and 60 native speakers, respectively.

3.3 Analysis and Results

Survey 1: Computer-Likeness Rating. Table 3 shows the mean 'computer-likeness'-ratings from 1 ('Certainly human') to 7 ('Certainly computer') for each response category. Clearly, the human responses are rated as more human-like than the computer-generated ones: our model did not generally mislead the participants. This may be due to the template-based response structure: over the course of the survey, human raters are likely to notice this structure and infer that such responses are computer-generated. To investigate whether such learning indeed occurs, a linear mixed-effects model was fitted, with predictor variables IS_COMP ($+1$:computer-generated, -1:human responses), POS (position of the trial in the survey, 0

Sent.	KB	Response
-8	No	Android is horrendous so I think that the writer is completely correct!!!
	Yes	Apple is horrendous so I feel that the author is not really right!!! iOS is horrendous as well.
-4	No	I think that the writer is mistaken because apple actually is unexceptional.
	Yes	I think that the author is wrong because Nokia is mediocre. Apple on the other hand is pretty good ...
0	No	The text is accurate. Apple is okay.
	Yes	Galaxy is okay so I think that the content is accurate. All-in-all samsung makes fantastic gadgets.
4	No	Android is pretty good so I feel that the author is right.
	Yes	Nokia is nice. The article is precise. Samsung on the other hand is fabulous...
8	No	Galaxy is great!!! The text is completely precise.
	Yes	Galaxy is awesome!!! The author is not completely correct. In fact I think that samsung makes awesome products.

Table 2: Responses generated by the system with or without a knowledge-base (KB), with different sentiment levels.

Response Type	Mean and CI
Human	3.33 ± 0.08
Computer (all)	4.49 ± 0.15
Computer ($-$KB)	4.66 ± 0.20
Computer ($+$KB)	4.32 ± 0.22

Table 3: Mean and 95% confidence interval of computer-likeness rating per response category. $\pm$KB indicates whether g_{base} or g_{kb} was used.

Factor	b	t	$P(b < 0)$
(intercept)	3.590		
IS_COMP	0.193	2.11	0.015
POS	0.069	4.76	0.000
IS_COMP $\times$ POS	0.085	6.27	0.000

Table 4: Computer-likeness rating regression results, comparing human to computer responses.

to 9), and the interaction between the two. Table 4 presents, for each factor in the regression analysis, the coefficient b and its t-statistic. The coefficient equals the increase in computer-likeness rating for each unit increase in the predictor variable. The t-statistic is indicative of how much variance in the ratings is accounted for by the predictor. We also obtained a probability distribution over each coefficient by Markov Chain Monte Carlo sampling using the `R` package `lme4` version 0.99 (Bates, 2005). From each coefficient's distribution, we estimate the posterior probability that b is negative, which quantifies the reliability of the effect.

The positive b value for POS shows that responses drift towards the 'computer'-end of the scale. More importantly, a positive interaction with IS_COMP indicates that the difference between human and computer responses becomes more noticeable as the survey progresses — the participants did learn to identify computer-generated responses. However, the positive coefficient for IS_COMP means that even at the very first trial, computer responses are considered to be more computer-like than human responses.

Factors Affecting Human-Likeness. Our finding that the identifiability of computer-generated responses cannot be fully attributed to their repetitiveness, raises the question: What makes a such a response more human-like? The results provide several insights into this matter.

First, the mean scores in Table 3 suggest that including a knowledge base increases the responses' human-likeness. To further investigate this, we performed a separate regression analysis, using only the data on computer-generated responses. This analysis also included predictors KB ($+1$: knowledge base included, -1: otherwise), SENT ($sentiment_t$, from -8 to $+8$), absolute value of SENT, and the interaction between KB and POS. As can be seen in Table 5, there is no reliable interaction between KB and POS: the effect of including the KB on the human-likeness of responses remained constant over the course of the survey.

Furthermore, we see evidence that responses with a more positive sentiment are considered more computer-like. The (only weakly reliable) negative effect of the absolute value of sentiment suggests that more extreme sentiments are considered more human-like. Apparently, people count on computer responses to be mildly positive, whereas human responses are expected to be more extreme, and extremely negative in particular.

Survey 2: Relevance Rating. The mean relevance scores in Table 6 reveal that a response is rated as more relevant to a snippet if it was actually a response to that snippet, rather than to a dif-

Factor	b	t	$P(b<0)$
(intercept)	4.022		
KB	−0.240	−2.13	0.987
POS	0.144	5.82	0.000
SENT	0.035	2.98	0.002
abs(SENT)	−0.041	−1.97	0.967
KB × POS	0.023	1.03	0.121

Table 5: Computer-likeness rating regression results, comparing systems with and without KB.

Factor	b	t	$P(b<0)$
(intercept)	3.861		
IS_COMP	−0.339	−7.10	1.000
SOURCE	0.824	16.80	0.000
IS_COMP × PRES	0.179	5.03	0.000

Table 7: Relevance ratings regression results, comparing human to computer responses.

Response Type	Source	Mean and CI
Human	this	4.85 ± 0.11
	other	3.56 ± 0.18
Computer (all)	this	4.52 ± 0.16
	other	2.52 ± 0.15
Computer (−KB)	this	4.53 ± 0.23
	other	2.46 ± 0.21
Computer (+KB)	this	4.51 ± 0.23
	other	2.58 ± 0.22

Table 6: Mean and 95% confidence interval of relevance rating per response category. 'Source' indicates whether the response is from the presented text snippet or a random other snippet. ±KB indicates whether g_{base} or g_{kb} was used.

Factor	b	t	$P(b<0)$
(intercept)	3.603		
KB	0.026	0.49	0.322
SOURCE	1.003	15.90	0.000
SENT	0.023	1.94	0.029
abs(SENT)	−0.017	−0.93	0.819
KB × SOURCE	−0.032	−0.61	0.731

Table 8: Relevance ratings regression results, comparing systems with and without KB.

ferent snippet. This reinforces our design choice to include input items referring specifically to the topic and sentiment of the author. However, human responses are considered more relevant than the computer-generated ones. This is confirmed by a reliably negative regression coefficient for IS_COMP (see regression results in Table 7).

The analysis included the binary factor SOURCE (+1 if the response came from the presented snippet, −1 if it came from a random article). We see a positive interaction between SOURCE and IS_COMP, indicating that presenting a response from a random article is more detrimental to relevance of computer-generated responses than that of the human responses. This is not surprising, as the computer-generated responses (unlike the human responses) always includes the article's topic.

When analyzing only data on computer-generated responses, and including predictors for agenda sentiment and for presence of the knowledge base, we see that including the KB does not affect response relevance (see Table 8). Also, there is no interaction between KB and SOURCE, that is, the effect of presenting a response from a different article does not differ between the models with and without the knowledge base. Possibly, responses are considered as more relevant if they have more positive sentiment, but the evidence for this is fairly weak.

4 Related and Future Work

In contrast to the vast amount of research on sentiment and topic analysis, as well as generation tasks in which the input is artificial or pre-defined, our system implements a full end-to-end cycle from natural language analysis to natural language generation with applications in social media and automated interaction in real-world settings.

The only two other studies on response generation in social media we know of are Ritter et al. (2011) and Hasegawa et al. (2013). Ritter's and Hasegawa's approaches differ from ours in their objective and their approach to generation. Specifically, Ritter's approach is based on machine translation, creating responses by directly re-using previous content. Their data-driven approach generates relevant, but not opinionated responses. In addition, both Ritter's and Hasegawa's systems respond to tweets, while our system analyzes and responds to complete articles. Hasegawa's approach is closer to ours in that it generates responses that are intended to elicit a specific emotion from the addressee. However, it still differs considerably in settings (dialogues versus online posting) and in the goal itself (eliciting emotion versus expressing opinion). Thus, we see these studies as comple-

mentary to ours in the realm of response generation in social media.

A natural contact point of our work with other existing work in social media analysis is the investigation of how a change in the implementation of individual components (e.g., topic inference or sentiment scoring) would affect the result of the overall generation. In particular, it would be interesting to test whether a novel mechanism for joint inference of topic/sentiment distributions could lead to improvement in the human-likeness of the generated responses.

The syntactic and semantic means of expression that we use are based on bare bone templates and fine-grained POS tags (Theune et al., 2001). These may potentially be expanded with different ways to express subject/object relations, relations between phrases, polarity of sentences, and so on. Additional approaches to generation can factor in such aspects, e.g., the template-based methods in Becker (2002) and Narayan et al. (2011), or grammar based methods, as in DeVault et al. (2008). Using more sophisticated generation methods with a rich grammatical backbone may combat the sensitivity to computer-generated response patterns as acquired by our human raters over time.

Furthermore, our result concerning the human-likeness of g_{kb} clearly demonstrates that semantic knowledge must be brought in to support better, and more human-like, response generation. Large-scale knowledge graphs such as Freebase support many semantic tasks (Jacobs, 1985), and can be used for providing richer context for automatically generating human-like responses.

From a theoretical viewpoint, the system will clearly benefit from rigorous analysis of human interaction in online media. Responses to user-generated content on the Internet share some linguistic characteristics in structure, length and manner of expression. Studying these features theoretically and then examining them empirically using a Turing-like evaluation as presented here can take us a big step in the direction of better generation, and also better understanding of the processes underlying human response generation.

This latter understanding may be complemented with insights into the causes, motivations and intricacies of human interaction in such environments, as studied by sociologists and psychologists. In particular, our preliminary interaction with colleagues from communication studies suggests that the present endeavor nicely complements that of "persuasive computing" (Fogg, 1998; Fogg, 2002), and we hope that this collaboration will lead to valuable synergies.

Finally, bridging the gap between the technical and the theoretical, it would be fascinating to test the responses in the context for which they are generated – social media. Generated texts may be posted as a response to the original article, or shared with a link of the original article, followed by measuring the responses to, and shares of, that response. Such real-world evaluation could indicate that generated responses are indeed believable and engaging, and may better simulate a Turing-like test in which machine-generated responses cannot be distinguished from human responses.

5 Conclusion

We presented a system for generating responses that are directly tied to responders' agendas and document content. To the best of our knowledge, this is the first system to generate subjective responses directly reflecting users' agendas. Our response generation architecture provides an easy-to-use and easy-to-extend solution encompassing a range of NLP and NLG techniques. We evaluated both the human-likeness and the relevance of the generated content, thereby empirically quantifying the efficacy of computer-generated responses compared head-to-head against human responses.

Generating concise, relevant, and opinionated responses that are also human-like is hard — it requires the integration of text-understanding and sentiment analysis, and it is also contingent on the expression of the agents' prior knowledge, reasons and motives. We suggest our architecture and evaluation method as a baseline for future research on generated content that would effectively pass a Turing-like test, and successfully convince humans of the authenticity of generated responses.[5]

Acknowledgments

We thank Yoav Francis for his contribution in early stages of this research and our anonymous reviewers for their insightful comments on an earlier draft.

[5] Our code, data, analysis scripts, implementation and raw data (computer and human responses) are publicly available via `www.tsarfaty.com/nlg-sd/`.

References

Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008. Finding high-quality content in social media. In *Proceedings of the international conference on Web search and web data mining*, pages 183–194. ACM.

Teresa M. Amabile. 1981. *Brilliant but Cruel: Perceptions of Negative Evaluators*. Washington, DC: ERIC Clearinghouse.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Douglas M. Bates. 2005. Fitting linear mixed models in R. *R News*, 5:27–30.

Tilman Becker. 2002. Practical, template-based natural language generation with TAG. In *Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022.

David M. Blei. 2012. Probabilistic topic models. *Commun. ACM*, 55(4):77–84.

Cristian Danescu-Niculescu-Mizil, Gueorgi Kossinets, Jon Kleinberg, and Lillian Lee. 2009. How opinions are received by online communities: A case study on amazon.com helpfulness votes. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 141–150, New York, NY, USA. ACM.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using Twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 241–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

David DeVault, David Traum, and Ron Artstein. 2008. Practical grammar-based NLG from examples. In *Proceedings of the Fifth International Natural Language Generation Conference*, INLG '08, pages 77–85, Stroudsburg, PA, USA. Association for Computational Linguistics.

Donghui Feng, Erin Shaw, Jihie Kim, and Eduard Hovy. 2006. An intelligent discussion-bot for answering student queries in threaded discussions. In *Proceedings of Intelligent User Interface (IUI-2006)*, pages 171–177.

B. J. Fogg. 1998. Persuasive computers: Perspectives and research directions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, pages 225–232, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

B. J. Fogg. 2002. Persuasive technology: Using computers to change what we think and do. *Ubiquity*, December.

Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*, ENLG '09, pages 90–93, Stroudsburg, PA, USA. Association for Computational Linguistics.

H. P. Grice. 1967. Logic and conversation. In H. P. Grice, editor, *Studies in the ways of words*, pages 22–40. Harvard University Press.

Michael Haenlein and Andreas M. Kaplan. 2009. Flagship brand stores within virtual worlds: The impact of virtual store exposure on real-life attitude toward the brand and purchase intent. *Recherche et Applications en Marketing (English Edition)*, 24(3):57–79.

Takayuki Hasegawa, Nobuhiro Kaji, Naoki Yoshinaga, and Masashi Toyoda. 2013. Predicting and eliciting addressee's emotion in online dialogue. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 964–972, Sofia, Bulgaria, August. Association for Computational Linguistics.

Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA. ACM.

Philip N. Howard, Aiden Duffy, Deen Freelon, Muzammil Hussain, Will Mari, and Marwa Mazaid. 2011. Opening closed regimes: What was the role of social media during the Arab spring? *Project on Information Technology and Political Islam.*

Paul S Jacobs. 1985. A knowledge-based approach to language production. Technical report, University of California at Berkeley, Berkeley, CA, USA.

Pulkit Kathuria. 2012. Sentiment Classification using WSD, Maximum Entropy and Naive Bayes Classifiers. *https://github.com/kevincobain2000/sentiment_classifier*. Visited March 2014.

Wiebke Lamer. 2012. Twitter and tyrants: New media and its effects on sovereignty in the Middle East. *Arab Media and Society*.

Marc Langheinrich and Günter Karjoth. 2011. Social networking and the risk to companies and institutions. *Information Security Technical Report. Special Issue: Identity Reconstruction and Theft*, pages 51–56.

Paul Mah. 2012. Tools to automate your customer service response on social media. *http://www.itbusinessedge.com/blogs/smb-tech/tools-to-automate-your-customer-service-response-on-social-media.html*. Visited August 2013.

Chris McConnell. 2012. When brands automate Twitter and Facebook responses I'll revolt. *http://dailytekk.com/2012/06/07/brands-automating-social-media/*. Visited August 2013.

Gilad Mishne. 2006. Multiple ranking strategies for opinion retrieval in blogs. In *Proceedings of the 15th Text Retrieval Conference*.

Kyoshi Mori, Adam Jatowt, and Mitsuru Ishizuka. 2003. Enhancing conversational flexibility in multimodal interactions with embodied lifelike agent. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pages 270–272, New York, NY, USA. ACM.

Mickey Nall. 2013. You can't automate social media engagement, argues PRSA's Mickey Nall. *http://www.prmoment.com/1359/you-cant-automate-social-media-engagement-argues-prsas-mickey-nall.aspx*. Visited August 2013.

Karthik Sankaran Narayan, Charles Lee Isbell Jr., and David L. Roberts. 2011. Dextor: Reduced effort authoring for template-based natural language generation. In Vadim Bulitko and Mark O. Riedl, editors, *Proceedings of the Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*. The AAAI Press.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In William W. Cohen and Samuel Gosling, editors, *ICWSM*. The AAAI Press.

Brendan O'Connor, Brandon M. Stewart, and Noah A. Smith. 2013. Learning to extract international relations from political context. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1104. The Association for Computer Linguistics.

Jeremiah Owyang. 2012. Brands Start Automating Social Media Responses on Facebook and Twitter. *http://techcrunch.com/2012/06/07/brands-start-automating-social-media-responses-on-facebook-and-twitter/*. Visited August 2013.

Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. 1998. Latent Semantic Indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '98, pages 159–168, New York, NY, USA. ACM.

Erik Qualman. 2012. *Socialnomics: How social media transforms the way we live and do business*. John Wiley & Sons, Hoboken, NJ, USA, 2nd edition.

Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Nat. Lang. Eng.*, 3(1):57–87.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 583–593, Stroudsburg, PA, USA. Association for Computational Linguistics.

M. Theune, E. Klabbers, J. R. De Pijper, E. Krahmer, and J. Odijk. 2001. From data to speech: A general approach. *Nat. Lang. Eng.*, 7(1):47–86.

Alan M. Turing. 1950. Computing machinery and intelligence. *Mind*, LIX:433–460.

Kees Van Deemter, Emiel Krahmer, and Mariët Theune. 2005. Real versus template-based natural language generation: A false opposition? *Comput. Linguist.*, 31(1):15–24.

Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. 2009. On the evolution of user interaction in Facebook. In *Proceedings of the 2nd ACM Workshop on Online Social Networks*, WOSN '09, pages 37–42, New York, NY, USA. ACM.

Tae Yano, William W. Cohen, and Noah A. Smith. 2009. Predicting response to political blog posts with topic models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 477–485, Stroudsburg, PA, USA. Association for Computational Linguistics.

תקציר

עבודה זה עוסקת ביצירת שפה טבעית בהקשר של מדיה חברתית. ככל שיותר ויותר תקשורת מתרחשת בצורה מקוונת, החשיבות של הבנה ותקשורת עם המשתמשים המקוונים נהיית חשובה לעסקים, ממשלות וארגונים נוספים. כמו כן, הרבה מהתקשורת הבין-אישית היום-יומית עוברת מערוצי התקשורת המסורתיים אל העולם הווירטואלי, דבר שמחזק את הצורך במחקר על תקשורת בין-אישית מקוונת.

בעבודה זאת אנו מכוונים ליצירת טקסטים טבעיים ואנושיים על ידי מחשב, טקסטים אשר מביעים דעה. בהסתמך על טכניקות מובילות בתחומן לניתוח והבנה של טקסטים, ותוך נתינת תשומת לב לניתוח של דעה (Sentiment) ונושאים מדוברים (Topic), אנו מגדירים ומיישמים מודל של משתמש מקוון ולאחר מכן בוחנים דרכים שונות ליצירה אוטומטית של טקסטים אשר נראים אנושיים ורלוונטיים.

ראשית בנינו מערכת אשר מסתמכת על תבניות, ליצירה אוטומטית של תגובות לכתבות מקוונות. המערכת משתמשת בתבניות דקדוק בעבודת-יד אשר מכילות שדות שמוחלפים בצורה דינאמית על ידי יצירה של ביטויי-יחוס (Referring Expressions) הרלוונטיים לתוכן המקוון.

אנו מציגים מבחן דמוי Turing להערכה של הטקסטים ומראים שהצלחנו להגיע לרמה קרובה באיכותה למשפטים שנוצרו ע"י משתמשים. באופן ספציפי, אנו מראים שהוספת ידע מן העולם (World Knowledge) לתגובות מעלה את איכות המשפטים ושתגובות עם דעה חיובית מיוחסות לתגובות ממוחשבות. בנוסף, אנו מראים בצורה אמפירית אפקט של למידה במהלך מבחני ההערכה – המעריכים האנושיים מצליחים ללמוד לזהות תגובות ממוחשבות ככל שמתקדם הניסוי. האפקט הזה נובע מהשונות הנמוכה המתאפשרת משימוש בתבניות.

בכדי להתגבר על בעיה זו של יצירת שפה בעזרת תבניות, ובפרט על בכדי להתגבר על השונות הנמוכה שאפשרית בצורת עבודה זו וההצמדה של התבניות לתחום מסוים, תכננו מערכת מונחת נתונים (Data-driven) אשר מבוססת על ארכיטקטורה מבוססת תחביר (Grammar). במסגרת זאת פיתחנו שלושה סוגי תחביר ליצירת שפה: 1) תחביר הסתברותי פשוט (PCFG), 2) תחביר לשוני (Lexicalized) אשר דומה לעבודה של Collins, 1997, ו-3) תחביר יחסי-התממשותי (Relational-Realizational) בהשראת העבודה של צרפתי ושות', 2008.

אנו משווים בין התחבירים השונים ע"י ביצוע של מבחן דמוי Turing באופן דומה למבחן לעיל, וכן מעריכים בצורה ממוחשבת את הקומפקטיות (Compactness), שטף (Fluency), והסכמת-דעה (Sentiment Agreement) של התגובות. מצאנו שתחביר יחסי-התממשותי הינו קומפקטי ומשיג תוצאות טובות יותר במודל שפה (Language Model) בעוד שתחביר לשוני מציג הסכמת-דעה טובה יותר במעט מאשר תחביר יחסי-התממשותי. בנוסף הראינו ששימוש במודל נושאי (Topic Model) ביצירת התגובות מאפשר לייצר משפטים רלוונטיים יותר. במבחני הערכה מקוונים קיבלנו תוצאות שונות במקצת כאשר התחביר לשוני מקבל הערכה טובה יותר של דימיון לתגובות אנושיות מאשר שני התחבירים האחרים.

התרומה של תיזה זאת היא אם כן מרובה: הצגנו משימה חדשה של יצירת שפה טבעית המביעה דעה וסיפקנו שתי תצורות כלליות ליצירה של משפטים דעתניים (Opinionated). בנוסף אנו משחררים לשימוש בסיס נתונים אשר מתאים להשראה (induction) של תחביר וכן הצגנו צורת הערכה חדשה לתחום זה. תוצאות התיזה מספקות תובנות חדשות בנוגע להבדלים בין שפה אשר נוצרת ע"י מחשב לזאת אשר נוצרת ע"י אנשים בתקווה שעבודה זאת תהווה השראה למחקר נוסף ולצורות מידול מתוחכמות ליצירת שפה דעתנית.

עבודה זו בוצעה בהדרכת של דר' רעות צרפתי מהמחלקה למתמטיקה ולמדעי המחשב באוניברסיטה הפתוחה, רעננה ופרופ' אריאל שמיר מבי"ס אפי ארזי למדעי המחשב, המרכז הבינתחומי, הרצליה.

המרכז הבינתחומי בהרצליה
בית-ספר אפי ארזי למדעי המחשב
התכנית לתואר שני (M.Sc.) - מסלול מחקרי

יצירת שפה טבעית דעתנית

מאת
תומר כגן

עבודת תיזה המוגשת כחלק מהדרישות לשם קבלת תואר מוסמך M.Sc. במסלול המחקרי בבית ספר אפי ארזי למדעי המחשב, המרכז הבינתחומי הרצליה

ספטמבר 2016