National Technical University of Athens
School of Electrical and Computer Engineering
Division of Signals, Control and Robotics

Tracking Intent and Personality Traits of Speakers in Spoken Dialogues Using Deep Learning

DIPLOMA THESIS

PINELOPI PAPALAMPIDI

Supervisor : Alexandros Potamianos
Associate Professor NTUA

Athens, June 2018

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Signals, Control and Robotics

Tracking Intent and Personality Traits of Speakers in Spoken Dialogues Using Deep Learning

DIPLOMA THESIS

PINELOPI PAPALAMPIDI

Supervisor : Alexandros Potamianos
Associate Professor NTUA

Approved by the examining committee on the June 25, 2018.

Alexandros Potamianos Associate Professor NTUA	Giorgos Stamou Associate Professor NTUA	Joakim Gustafsson Professor KTH

Athens, June 2018

Pinelopi Papalampidi
Electrical and Computer Engineer

Abstract

In this work, we address two important issues related to the improvement of the performance of dialogue systems in terms of human-machine interaction. The first issue, dialogue act classification, is typically the most fundamental in dialogue processing. It allows the dialogue systems to select the most appropriate response from a set of possible generated replies based on the communicative intentions of the user at that moment. The second issue, personality recognition, aims at the further improvement of the dialogue systems in order to successfully adapt to an individual user's behavior. This adaptation assists in developing personalized dialogue systems.

Dialogue acts are considered as the minimal linguistic units of communicative intentions in terms of dialogues. However, the utilized annotation schemes for dialogue act recognition are dependent on the task and domain of each dataset. In this work, we propose a method for successfully recognizing the intentions of an interlocutor independently from the task of the dialogue and the granularity of the dialogue act tag-set utilized. For this purpose, we implement a deep learning model for recognizing the dialogue acts based on the semantic representation of the current utterance as well as the history of the dialogue so far. In this architecture, we propose the expansion of generic word embeddings, that are used for initializing the embedding layer of the respective neural network, with dialogue act-specific information. Specifically, first we automatically extract a set of keywords that is considered representative for each dialogue act tag, forming a semantical subspace for each tag. Next, we compute the semantic similarity between each word and each dialogue act tag, by computing the similarity between each word and the respective set of keywords. Finally, we concatenate the generic word embeddings with the custom word vectors and fed them to our neural model. We evaluate our approach in a dataset commonly used for dialogue act classification and achieve results comparable with the state-of-the-art.

Next, we address the problem of automatic personality recognition. Personality traits are described with the Big Five model, derived from psychological studies. This model is considered sufficient for outlining the human personality across different languages and cultures. In this work, we adopt the hypothesis that the personality traits are nevertheless dependent on the context of a given situation and hence, related to the behavioral and emotional state as well as the intention of the individual. In fact, we first introduce the relevance of intent recognition to the personality recognition problem. We aim at incorporating emotion and intent knowledge to the automatic personality recognition problem. We propose a novel adaptation of two well-known neural transfer learning methods for incorporating sentence-level emotion and intent information to document-level personality recognition. Our models are based on hierarchical attention networks. First, we train a model on a sentence-level source task (i.e. emotion, intent or both via multi-task learning). Next, we utilize the encoder of the pre-trained model for fine-tuning on the target task or as a sentence-level feature extractor. The suggested approach achieves state-of-the-art results in two personality datasets. We also evaluate the incorporation of lexicon-based psycholinguistic features to our model, as already suggested in the literature. Finally, we conduct an analysis on the contribution of different information sources to the problem and validate our initial assumption.

Key words

dialogue act, deep learning, recurrent neural networks, semantic similarity, personality recognition, hierarchical attention network, transfer learning, dialogue systems

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Alexandros Potamianos for his guidance during the last years that I am conducting my diploma thesis. His valuable advices helped me in improving my research and publishing my work.

Next, I would like to thank Prof. Giorgos Stamou and Prof. Joakim Gustafsson for being a part of my thesis committee and my colleagues in SLP-NTUA lab for both assisting me in my research work (special thanks to Christos Baziotis for the over-the-limits effort to catch the EMNLP 2018 deadline) and sharing inspirational talks with me.

Finally, I deeply thank my family for supporting and helping me in any way possible in order to achieve my goals for the last 24 years of my life. I also thank my friends, with whom I shared both good and stressful moments all those years.

Pinelopi Papalampidi,

Athens, June 25, 2018

To my family.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Human communication usually aims at agreement on some situation definition. However, often this agreement is not immediate and as a result a dialogue is needed in order for the interlocutors to reach a decision. During the process of the dialogue, the interlocutors establish a common conceptual ground, where they agree upon shared beliefs, mutual knowledge, joint goals and joint intentions and are able to recognize each other's communicative motives. In order to recognize a speaker's communicative motives, the Dialogue Acts (DAs), which are considered as the minimum unit of linguistic communication and are directly connected with the speaker's communicative intentions, should be tracked as the dialogue evolves. Moreover, the behavior of each individual during the interaction is conditioned on the behavior and traits of his/her interlocutor. Hence, being able to recognize an individual's personality traits (e.g. Extraversion, Agreeableness, Openness according to the Big Five model for analyzing human personality) can be helpful for the establishment of a common conceptual ground in human communication. The main objectives of the thesis is intent and personality recognition, since these are two of the fundamental elements in any kind of communication, either human-human or human-machine interaction.

Based on the rules that govern the human communication, a major processing step for the improvement of the human-machine interaction in the context of Spoken Dialogue Systems (SDS) is DA classification. Recognizing the DAs can assist in the understanding of the user input. Typically, this is implemented as the assignment of tags to user utterances that (lexically) describe the respective acts. Moreover, the ability of a SDS to identify and adapt to the individual's personality can improve the degree of engagement of people when interacting with a machine, as well as the possibility of reaching an agreement in the end of the dialogue. Furthermore, specific personality traits can be adopted by the SDS in order to generate responses conditioned on these traits.

Based on the above definition and analysis of human communication, the main goal of this thesis is to improve the human-machine interaction (e.g. SDS) by equipping the "machine" with fundamental abilities in communication: recognizing the human communicative motives and personality traits. Specifically, we first address the task of automatic intent recognition as it is considered the basic element of any human communication. Next, we move on to the problem of automatic personality recognition. In this task, we adapt the hypothesis, sustained in the literature, that personality recognition is conditioned on the context of the situation under examination, for example the behavioral and emotional state as well as the intentions of the individual. In fact, to the best of our knowledge, we first introduce the intent recognition as a task related to personality recognition.

1.1 Intent Recognition

The first research question of the thesis is the recognition of the intentions of the interlocutors as the dialogue proceeds, since DA recognition is traditionally the first step in dialogue processing.

A challenge in this domain is the definition of a taxonomy of DAs that can be generalized and used across different domains and dialogues of different nature. Although attempts have been made for the establishment of a global annotation scheme in DA classification, there are still limited data annotated with such a scheme. Consequently, another challenge is the implementation of a model for DA classification that can successfully predict the DAs in the context of a dialogue, independently

from the task and domain of the dialogue as well as the DA taxonomy utilized for the annotation of the specific dataset.

Previous approaches are based on feature engineering in order to correctly identify the DAs in the context of dialogue. They typically extract lexical, syntactical and semantic information related to the emotions of the speakers as well as features related to the history of the dialogue (e.g. DAs of previous utterances). Most recent approaches have exploited Deep Neural Networks (DNNs) in order to capture the information concealed in both utterance and dialogue level. For this purpose several architectures, including Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) have been suggested in the literature. However, such models miss semantic information related to the specific problem of DA recognition, presenting as a result a variance in their performance depending on the different utilized datasets and their respective set of DAs used for the annotation.

In this work, we propose the expansion of generic word embeddings with word vectors that denote the semantic similarity of each word with each one DA tag. These word embeddings are computed based on semantic subspaces corresponding to each DA tag, constructed based on the automatic extraction of keywords that are considered representative of these tags. Next, the concatenation of the generic word embeddings with our custom word vectors are fed to a DNN for the prediction of the DA for each utterance of each dialogue. The augmented word embeddings that we propose are automatically computed and able to adjust to datasets of different domains and DA annotation schemes of different granularity. In our experiments, we utilize a well-known dataset, commonly used for this purpose in the literature, for evaluating our approach and we achieve results that are comparable with the respective state-of-the-art.

1.2 Personality Recognition

The second research question addressed in the thesis, that aims at the improvement of the performance of SDSs is the recognition of personality traits of an individual. The automatic detection of an individual's personality traits has many applications in both real life and other Artificial Intelligence (AI) problems.

In recommendation systems, clustering the personality traits of users in large groups can optimize the recommendations addressed to each person and benefit from the preferences of people with similar personality. In mental health diagnosis, certain diagnoses correlate with certain personality traits. However, the applications of personality recognition are not restricted to such problems. The performance of Dialogue Systems (DS) can also be boosted if the DS is able to identify and adapt to the individual's personality. Moreover, specific personality traits can be adopted by the DS in order to generate responses conditioned on these traits.

The dominant model derived from psychology for describing the human personality is the Big Five model (see Section). This system is considered sufficient for describing the human personality in literature. Specifically, as sustained by [1], "if a large number of rating scales is used and if the scope of the scales is very broad, the domain of personality descriptors is almost completely accounted for by five robust factors". The core components of the personality system are designated as "basic tendencies", "characteristic adaptations" and the "self-concept" [2]. Characteristic adaptations, however, are influenced by external influences, such as the situation or the context under which the personality is examined. Therefore, although the personality traits are considered stable across time in some degree, they are, nevertheless, associated with the circumstances under which the behaviors of the individual are examined [3]. This theoretical assumption poses the challenge of incorporating information sources related to personality recognition to the target task. Moreover, another significant challenge in automatic personality recognition is the limitation presented in annotated data. Since labeling in personality recognition is time consuming process, where annotations are made in individual-level, the available datasets for training and testing automatic personality recognition methods contain a limited number of samples.

Previous successful approaches on personality recognition leverage lexical information using both

traditional and deep learning architectures [4, 5, 6]. In particular, for the problem of document-level personality recognition (i.e. when the prediction of an individual's personality traits is based on an extended document, such as a dialogue, monologue or essay), State-of-the-art models also incorporate psycholinguistic and affective features, typically extracted from knowledge bases or lexicons, at the document level.

In the context of the thesis, we propose a deep learning approach free of hand-crafted features and lexicons. Specifically, based on the hypothesis that personality recognition is conditioned on the context of a given situation, we propose a novel adaptation of two well-known Transfer Learning (TL) methods for incorporating information related to personality recognition (i.e. intent, emotion, psycholinguistics) to the target problem (i.e. personality recognition). We transfer knowledge from sentence-level tasks to the document-level problem and further, we conduct an analysis on the contribution of different information sources to recognizing each personality trait of authors and speakers. We achieve state-of-the-art results under most experimental results and conclude on which information source is more dominant in recognizing each personality trait for different personality dataset.

1.3 Thesis Outline

The thesis is structured as follows. In Chapter 2 the background theory is presented. Specifically, the concepts of intentionality, speech act and DA are explained in detail and the various taxonomies that have been constructed for their analysis are presented. Next, the theories for the analysis of the personality system are introduced. An overview of the basic techniques and trends in Natural Language Processing (NLP) and Machine Learning (ML) follows. The last part of Chapter 2 is dedicated to the introduction in Deep Learning (DL) with a special focus on Recurrent Neural Networks (RNNs), Bidirectional Long Short-Term Memory (BiLSTM) networks and Transfer Learning (TL). After presenting the background theory, in Chapters 3 and 4, the proposed approaches utilizing in order to address the two research questions are presented: Intent classification and personality recognition. Specifically, in Chapter 3, the methodology and models implemented for DA representation and classification is presented alongside with the respective experimental results and conclusions. Next, in Chapter 4, an approach that takes advantage of semantic information derived from intent and emotion recognition via TL is proposed for personality recognition. The results of the conducted experiments as well as the conclusions drawn are presented. Finally, Chapter 5 generally concludes and proposes some ideas as future work.

Chapter 2

Background Theory

2.1 Intent Recognition

2.1.1 Psychological Analysis of Intentionality

The first part of the analysis of intentionality is the definition of human communication. Human communication is a fundamentally cooperative enterprise, operating most naturally and smoothly within the context of

- mutually assumed common conceptual ground
- mutually assumed cooperative communicative motives

Human cooperation is structured by what some modern philosophers of action call shared intentionality or ”we” intentionality [7]. In general, shared intentionality is the necessary element for engaging in uniquely human forms of collaborative activity in which a plural subject ”we” is involved: joint goals, joint intentions, mutual knowledge, shared beliefs. Cooperative communication thus arose as a way of coordinating these collaborative activities more efficiently, first inheriting and then helping to build further a common psychological infrastructure of shared intentionality.

In human communication, according to Tomasello [8] , there are three specific hypotheses:

- Human cooperative communication emerged first in evolution (and emerges first in ontogeny) in the natural, spontaneous gestures of pointing and pantomiming.
- Human cooperative communication rests crucially on a psychological infrastructure of shared intentionality, which originated evolutionarily in support of collaborative activities, and which comprises most importantly:
  - social-cognitive skills for creating with others joint intentions and joint attention (and other forms of common conceptual ground)
  - prosocial motivations (and even norms) for helping and sharing with others
- Conventional communication, as embodied in one or another human language, is possible only when participants already possess:
  - natural gestures and their shared intentionality infrastructure
  - skills of cultural learning and imitation for creating and passing along jointly understood communicative conventions and constructions

The meaning of the word *’intentionality’* should not be confused with the ordinary meaning of the word ’intention’. As the Latin etymology of *’intentionality’* indicates, the relevant idea of the directness or tension arises from pointing towards or attending to some target. In medieval logic and philosophy, the Latin word *intentio* was used for what contemporary philosophers and logicians nowadays call a ’concept’ or an ’intension’: something that can be both true of non-mental things and properties-things and properties lying outside the mind - and present to the mind. On the assumption that a concept is itself something mental, an intentio may also be true of mental things. Although the meaning of

the word 'intentionality' in contemporary philosophy is related to the meanings of such words as 'intension' and 'intention', nonetheless it ought not to be confused with either of them.

In contemporary English, *'intensional'* and *'intensionality'* mean *'non-extensional'* and *'non-extensionality'*, where both extensionality and intensionality are logical features of words and sentences. On the other hand, intention and intending are specific states of mind that unlike beliefs, judgments, hopes, desires or fears, play a distinctive role in the etiology of actions. By contrast, intentionality is a pervasive feature of many different mental states: beliefs, hopes, judgments, intentions, love and hatred all exhibit intentionality. In fact, Brentano held that intentionality is the hallmark of the mental [9].

Furthermore, it is worthwhile to distinguish between levels of intentionality. Many of an individual's psychological states with intentionality (e.g. beliefs) are about non-mental things, properties and states of affairs. Many are also about another's psychological states (e.g. another's beliefs). Beliefs about others' beliefs display what is known as *'higher-order intentionality'*.

It is a fact, thus, that some mental states exhibit intentionality. This would contribute to explaining why an individual's behavior consists in a pair whose coordinates are respectively an internal state and a bodily movement such that the former causes the latter. Dretske [10] has espoused a componential view of behavior according to which an individual's behavior is not to be identified, as on the functionalist conception, with his or her bodily movement: behavior is the process whereby some of the individual's bodily movement is caused by one of his or her internal state. When the internal state has intentionality, the individual's behavior is intentional. On the componential view of behavior, the intentionality of an individual's mental state is not relevant to the causation of a particular movement at time t. Rather, it is relevant to why types of movements are regularly caused by types of intentional states.

2.1.2 Definition of Communicative Acts

Communication is a process aimed at agreement on some situation definition. When the agreement is not immediate, a discussion is needed to resolve the points of disagreement using argumentation. In other words, the notion of the function of a communication action is defined in terms of the flow of information between the speaker and the addressee. This is why argumentation theory is important in order to understand how the communication is going to evolve. Argumentation theory is the interdisciplinary study of how conclusions can be reached through logical reasoning. That is claims based on premises. It includes the art and sciences of civil debate, dialogue, logic and procedural rules in both artificial and real world settings. The results of this theory in combination with the Language/Action Perspective (LAP) approaches, provides us of a model, which is further developed in confrontation with the well-known Issue Based Information System (IBIS) IBIS framework of Conklin [11], according to which there is no perfect solution for a wicked problem. Based on the LAP approach, language is considered as the primary dimension of human cooperative activity. The final model is a so-called-3-box model that is proposed as an extension of the Transaction Process Model (TPM) of Reijswoud [12] and its most important actions are adding claims, arguments, advantages and disadvantages. The model of Reijswoud is the following, where the CAs are communicative actions. For example, CA1 is a request and CA2 the acceptance of the request. The alternative to CA2 is CA5 (request justification), by means of which the Hearer moves into the Discussion and Failure layer.

So, as observed in Figure 2.1, the TPM has three communication layers, the success layer, the discussion and failure layer and the discourse layer, with which the background norms and rules can be revitalized or adapted towards agreement. In the LAP, the semantics of conversations is usually described in terms of speech acts and the effects of these in terms of beliefs, intentions and obligations [13].

However, the communication actions interact with common ground [14, 15]. Specifically, the semantics of communicative actions is given in terms of claims, and these claims get their support from the shared background in the community. The effects of the communicative actions appeal to the common ground as well: once a claim is conceded, its content becomes mutually accepted. As a

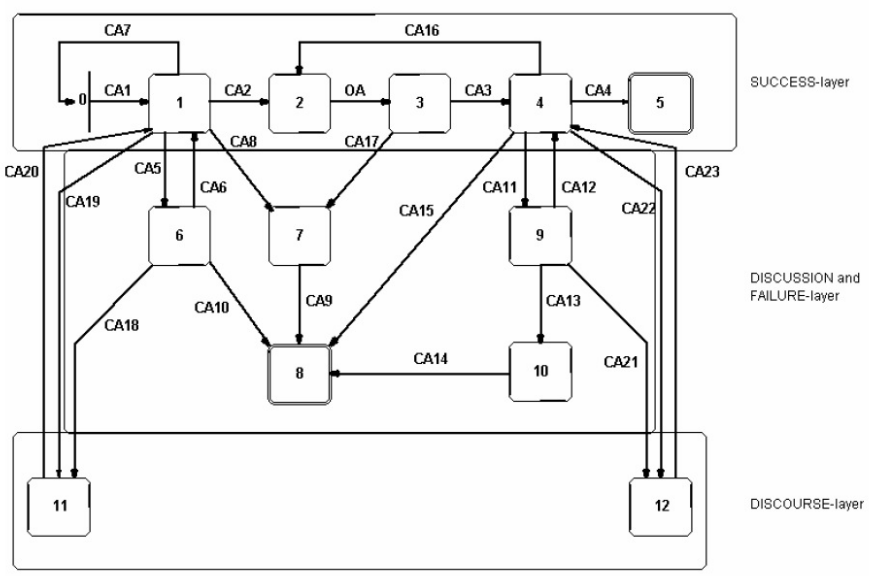


Figure 2.1: Overview of the Transaction Process Model

result, argumentation, that takes place during communication, can be viewed as the process of building a bridge between what is agreed upon in the community (or relationship) in the form of common ground (shared norms, in the sense of Stamper [16]) and the situation at hand (”grounding”).

The challenge related to communication acts is the need of immediate process and response of the system to what the speaker tells. Dynamic predicate logic by Jeroen Groenendijk and Martin Stokhof [17] is helpful for the online semantic interpretation. This system comes as a result of the formulation and investigation of a dynamic semantic interpretation of the language of first-order predicate logic and is intended as a first step towards a composition, non-representational theory of discourse semantics.

2.1.3 Definition of Speech Acts & Dialogue Acts

The idea that human conversation contains speech acts originates from socio-linguistic theorists [18, 19]. Dialogue Act (DA) theory suggests that humans not only communicate factual information within natural language utterances, but also they often express underlying intended action.

The first step of dialogue processing is the DA, the assignment of a functional tag to user’s input aiming at the representation of the communicative intentions behind each utterance. However, according to Inference-Based approach, the intentions can be also based on the analysis of the complete dialogue, rather than a single utterance so as to find a consistent semantic representation capturing the meaning of the dialogue. Either way this first step is crucial for an automated system so as to be able to produce an appropriate response. There is a wide range of uses of dialogue acts, including representations of the pragmatic meaning of utterances in dialogue theories [20, 21, 22, 23], building blocks of dialogue [24, 25], labels for corpus annotation [26, 27], agent communication languages [28, 29], object of analysis in dialogue systems [30, 31], and element of logical theory of rational interaction [32]. However, there is still difficulty in creating a taxonomy of dialogue acts that can be understood and used by researchers other than the taxonomy designers. This difficulty derives from the different interpretations that researchers assign to the different dialogue act taxonomies. This kind of confusion has led some (e.g. [33, 29]) to propose standard theories that could be well-defined, understood and used across groups, while others (e.g. [34, 35]) prefer to treat dialogue act identification as of only secondary importance, as a derived concept within a more general theory of rational interaction, using concepts as primitives. Another important issue is the recognition of the dialogue acts in a dialogue between a system and a person. The accurate recognition of the dialogue acts from a dialogue system demands a well-designed language understanding system. In order to design such a system, the fol-

lowing aspects should be taken into consideration: syntactics, i.e. relations between utterances and the structure of the sentences, semantics, i.e. anaphora and ellipsis, and pragmatics, i.e. the analysis of information-exchange dialogues as consisting of communicative actions.

The question that now arises is how actions are used in practice for the implementation of a conversational system. Actions are considered as transitions from states to states while dialogue acts as special cases of actions. Theories of action proposed by Artificial Intelligence (AI) researches generally associate several sets with actions: a set of effects (constraints on the resulting state), a set of pre-conditions (constraints on the initial state) and decompositions (subactions that performed together constitute the action).

Based on the above definition of action, the aspects of the situation that are relevant as potential conditions for defining types of dialogue act performance and the ones that are directly affected should be determined. The conditions and effects of dialogue acts that is used are the following:

- the notion of dialogue state, as encoded as state in dialogue grammar [24, 36].
- the mental states (e.g. belief, intention) of the speaker and the addresse [37, 38], which is the most popular in the planning approach
- the social obligations and commitments undertaken by the dialogue participants [39, 23, 37]

2.1.4 Applications & Taxonomies of DAs

Applications where dialogue systems are required, range from simple tasks such as operating a home device or booking a flight to more complex tasks such as controlling a smart-room or managing the road traffic. Due to the complexity of the management of language interfaces and their strong dependence on the interaction context each application or a set of applications requires the development of a specific model. That is the reason why most current prototyping methods are limited to the development of dialogue systems working on a single application or a small set of applications. Dialogue prototyping, therefore represents a significant part in the development process of interactive system, especially for the ones with a vocal interface: there is a strong need for an efficient Rapid Dialogue Prototype Methology (RPDM) [40]. The RDPM contains five main steps: 1. producing a task model for the targeted application 2. deriving an initial dialogue model from the obtained task model 3. carrying out a Wizard-of-Oz experiment to improve the initial dialogue model 4. carrying out an internal field test to further refine the dialogue model (reformulation of system messages, improved feedback, etc.) and to validate the evaluation procedure (coherence, understandability) 5. carrying out an external filed test to evaluate the final dialogue model according to the evaluation procedure defined during the internal field test.

There is a large range of applications of DAs that are based on automatic dialogue acts detection. Among them the most important ones are: dialogue systems, machine translation, Automatic Speech Recognition (ASR), topic identification and animation of talking head. In dialogue systems, DAs can be used to recognize the intention of the user, for instance when the user is requesting some information and is waiting for it, or when the system is trying to interpret the feedback from the user. An example of dialogue management system that uses DA classification is the VERBMOBIL system.

In machine translation, dialogue acts can be useful to choose the best solution when several translations are available. In particular, the grammatical form of an utterance may depend on its intention. Automatic detection of dialogue acts can be used in ASR to increase the word recognition accuracy. A talking head is a model of human head that reproduces the speech of a speaker in real-time. It may also render facial expressions that are relevant to the current state of the discourse. Exploiting DA recognition in this context might make the animation more natural, for example by raising the eyebrows when a question is asked. Another easier option is to show this complementary information with symbols and colors near the head.

Due to the differences in the requirements of these applications as far as the necessary DA tag-set is concerned, the DA tag-set definition is an important but difficult step. That is because it results form a compromise between three conflicting requirements:

- The DA labels should be generic enough to be useful for different tasks, or at least robust to the unpredictable variability and evolution of the target application.
- The DA labels must be specific enough to encode detailed and exploitable characteristics of the target task.
- The DA labels must be clear and easily separable, in order to maximize the agreement between human labelers.

Many different DA tag-sets can be found in the literature. Recently, a few of them seem to emerge as a common baseline, from which application-specific DA tags are derived. These are the Dialogue Act Markup in Several Layers (DAMSL) [41], the Switchboard SWBD-DAMSL [42], the Meeting Recorder [43], the VERBMOBIL [44] and the Map-Task [45] DAs tag-sets.

DAMSL was initially designed to be universal. Its annotation scheme is composed of four levels (or dimensions): communicative status, information level, forward looking functions and backward looking functions. Generally, these dimensions are considered as orthogonal and it shall be possible to build examples for any possible combination of them. The communicative status states whether the utterance is uninterpretable, abandoned or is a self-talk. This feature is not used for most utterances. The information level provides an abstract characterization of the content of the utterance. It is composed of four categories: task, task-management, communication-management and other-level. The forward looking functions are organized into a taxonomy, in a similar way as actions in traditional speech act theory. The backward looking functions show the relationship between the current utterance and the previous dialogue acts, such as accepting a proposal or answering the question. SWBD-DAMSL is the adaptation of DAMSL to the domain of the telephone conversations. Most of the SWBD-DAMSL labels actually correspond to DAMSL labels.

The Meeting Recorder DA (MRDA) tag-set is based on the SWBD-DAMSL taxonomy. The MRDA corpus contains about 72 hours of naturally occurring multi-party meetings manually-labels with DAs and adjacency pairs. Meetings involve regions of high speaker overlap, affective variation, complicated interaction structures, abandoned or interrupted utterances and other interesting turn-talking and discourse-level phenomena.

The DA hierarchy in VERBMOBIL is organized as a decision tree. This structure is chosen to facilitate the annotation process and to clarify relationships between different DAs.

The DA tags in the Map Task corpus are structured into three levels, the highest modeling transactions, where each transaction accomplishes one major step in the speakers' plan. Transactions are then composed of conversational games, which model the regularity between questions/answers, statements/deny or acceptance, and so on. Games are finally made up of conversational moves, which classify different kinds of games according to their purposes.

2.1.5 Modern Applications of Dialogue Systems

Task-oriented dialog agents are designed for a particular task and set up to have short conversations (from as little as a single interaction to perhaps half-a-dozen interactions) to get information from the user to help complete the task. These include the digital assistants that are now on every cellphone or on home controllers (Siri, Cortana, Alexa, Google Now/Home, etc.) whose dialog agents can give travel directions, control home appliances, find restaurants, or help make phone calls or send texts. Companies deploy goal-based conversational agents on their websites to help customers answer questions or address problems. Conversational agents play an important role as an interface to robots. And they even have applications for social good. DoNotPay is a "robot lawyer" that helps people challenge incorrect parking fines, apply for emergency housing, or claim asylum if they are refugees. **Chatbots** are systems designed for extended conversations, set up to mimic the unstructured conversational or 'chats' characteristic of human-human interaction, rather than focused on a particular task like booking plane flights. These systems often have an entertainment value, such as Microsoft's 'XioaIce' system, which chats with people on text messaging platforms. Chatbots are also often attempts to pass

various forms of the Turing test. Yet starting from the very first system, ELIZA [46], chatbots have also been used for practical purposes, such as testing theories of psychological counseling. Note that the word 'chatbot' is often used in the media and in industry as a synonym for conversational agent.

2.2 Personality Recognition

2.2.1 Theories of Personality

There is a variation in defining the human personality. According to Allport [47], "personality is the dynamic organization within the individual of those psychophysical systems that determine his characteristics behavior and thought", whereas Weinbeirg and Gould [48] sustain that personality consists of "the characteristics or blend of characteristics that make a person unique". Both of these definitions emphasize the uniqueness of the individual and consequently adopt an idiographic view. The idiographic view assumes that each person has a unique psychological structure and that some traits are possessed by only one person; and that there are times when it is impossible to compare one person with others. It tends to use case studies for information gathering. The nomothetic view [49] , on the other hand, emphasizes comparability among individuals. This viewpoint sees traits as having the same psychological meaning in everyone. This approach tends to use self-report personality questions, factor analysis, etc. People differ in their positions along a continuum in the same set of traits.

Trait theories of personality imply personality is biologically based, whereas state theories such as Bandura's [50] emphasize the role of nurture and environmental influence. Sigmund Freud's psychodynamic theory of personality assumes there is an interaction between nature (innate instincts) and nurture (parental influences).

Freud's Theory: Tripartite Theory of Personality

According to Freud, personality involves several factors:

- Instinctual drives
- Unconscious processes
- Early childhood influences

Personality development depends on the interplay of instinct and environment during the first five years of life. Parental behavior is crucial to normal and abnormal development.

Freud [51] saw the personality structured into three parts (i.e., tripartite) all developing at different stages in our lives and seperate from the brain:

- The **id** is the primitive and instinctive component of personality. It consists of all the inherited (i.e., biological) components of personality, including the sex (life) instinct and aggressive (death) instinct.
- The **ego** develops in order to mediate between the unrealistic id and the external real world (like a referee). It is the decision-making component of personality.
- The **superego** incorporates the values and morals of society which are learned from one's parents and others. It is similar to a conscience, which can punish the ego through causing feelings of guilt.

Trait Approach to Personality

This approach assumes behavior is determined by relatively stable traits which are the fundamental units of one's personality. Traits predispose one to act in a certain way, regardless of the situation. This means that traits should remain consistent across situations and over time, but may vary between individuals. It is presumed that individuals differ in their traits due to genetic differences. These theories are sometimes referred to a psychometric theories, because of their emphasis on measuring personality by using psychometric tests. Trait scores are continuous (quantitative) variables. A person is given numeric score to indicate how much of a trait the they possess. All other theories developed (see following subsections) are based on this assumption.

Eysenck's Personality Theory

Eysenck [52, 53, 54] proposed a theory of personality based on biological factors, arguing that individuals inherit a type of nervous system that affects their ability to learn and adapt to the environment. During 1940s Eysenck was working at the Maudsley psychiatric hospital in London. His job was to make an initial assessment of each patient before their mental disorder was diagnosed by a psychiatrist. Through this position, he compiled a battery of questions about behavior, which he later applied to 700 soldiers who were being treated for neurotic disorders at the hospital (Eysenck (1947). He found that the soldiers' answers seemed to link naturally with one another, suggesting that there were a number of different personality traits which were being revealed by the soldier's answers. He called these first-order personality traits He used a technique called factor analysis. This technique reduces behavior to a number of factors which can be grouped together under separate headings, called dimensions. Eysenck [55] found that their behavior could be represented by two dimensions: Introversion / Extroversion (E); Neuroticism / Stability (N). Eysenck called these second-order personality traits. Each aspect of personality (extraversion, neuroticism and psychoticism) can be traced back to a different biological cause. Personality is dependent on the balance between excitation and inhibition process of the autonomic nervous system (ANS).

- **Extraversion/introversion**: Extraverts are sociable and crave excitement and change, and thus can become bored easily. They tend to be carefree, optimistic and impulsive. They are more likely to take risks and be thrill seekers. Eysenck argues that this is because they inherit an under aroused nervous system and so seek stimulation to restore the level of optimum stimulation. Introverts, on the other hand, lie at the other end of this scale, being quiet and reserved. They are already over-aroused and shun sensation and stimulation. Introverts are reserved, plan their actions and control their emotions. They tend to be serious, reliable and pessimistic.

- **Neuroticism/stability**: A person's level of neuroticism is determined by the reactivity of their sympathetic nervous system. A stable person's nervous system will generally be less reactive to stressful situations, remaining calm and level headed. Someone high in neuroticism on the other hand will be much more unstable, and prone to overreacting to stimuli and may be quick to worry, anger or fear. They are overly emotional and find it difficult to calm down once upset. Neurotic individuals have an ANS that responds quickly to stress.

- **Psychoticism/normality**: Eysenck [56] later added a third trait / dimension - Psychoticism – e.g., lacking in empathy, cruel, a loner, aggressive and troublesome. This has been related to high levels of testosterone. The higher the testosterone, the higher the level of psychoticism, with low levels related to more normal balanced behaviour.

According to Eysenck, the two dimensions of neuroticism (stable vs. unstable) and introversion-extroversion combine to form a variety of personality characteristics.

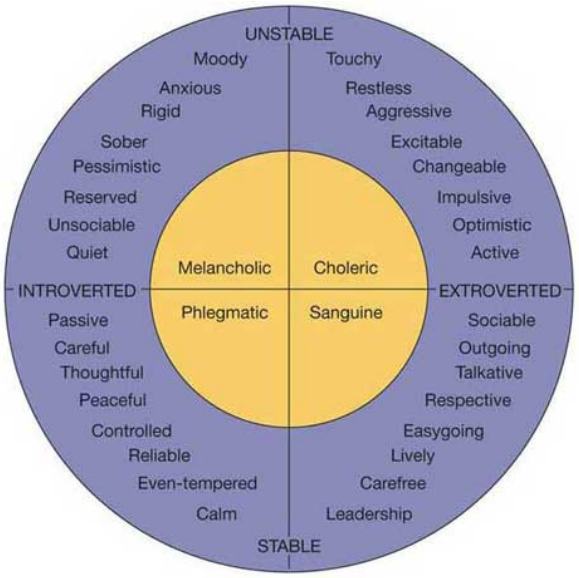


Figure 2.2: Eysenck's distribution of personality characteristics

Cattell's 16PF Trait Theory

Cattell [57] disagreed with Eysenck's view that personality can be understood by looking at only two or three dimensions of behavior. Instead, he argued that that is was necessary to look at a much larger number of traits in order to get a complete picture of someone's personality. Whereas Eysenck based his theory based on the responses of hospitalized servicemen, Cattell collected data from a range of people through three different sources of data.

- L-data: this is life record data such as school grades, absence from work, etc.
- Q-data: this was a questionnaire designed to rate an individual's personality (known as the 16PF).
- T-data: this is data from objective tests designed to 'tap' into a personality construct.

Cattell analyzed the T-data and Q-data using a mathematical technique called factor analysis to look at which types of behavior tended to be grouped together in the same people. He identified 16 personality traits / factors common to all people. Cattell made a distinction between source and surface traits. Surface traits are very obvious and can be easily identified by other people, whereas source traits are less visible to other people and appear to underlie several different aspects of behavior. Cattell regarded source traits are more important in describing personality than surface traits.

Allport's Trait Theory

Allport's theory of personality emphasizes the uniqueness of the individual and the internal cognitive and motivational processes that influence behavior. For example, intelligence, temperament, habits, skills, attitudes, and traits. Allport [58] believes that personality is biologically determined at birth, and shaped by a person's environmental experience. According to Allport, personality trains can be categorized into three general levels:

- **Cardinal Traits**: dominate the entirety of a person's life.
- **Central Traits**: are general characteristics used to describe another person (e.g. kind, sincere, cool and jolly).
- **Secondary Traits**: are those that only come out under certain situations.

Factor	Low Score	High Score
Warmth	cold, selfish	supportive, comforting
Intellect	Instinctive, unstable	cerebral, analytical
Emotional Stability	Irritable, moody	level headed, calm
Aggressiveness	Modest, docile	controlling, tough
Liveliness	somber, restrained	wild, fun loving
Dutifulness	untraditional, rebellious	conformity, traditional
Social Assertiveness	shy, withdrawn	uninhibited, bold
Sensitivity	coarse, tough	touchy, soft
Paranoia	trusting, easy going	wary, suspicious
Abstractness	practical, regular	strange, imaginative
Introversion	open, friendly	private, quiet
Anxiety	confident, self-assured	fearful, self-doubting
Open-mindedness	close-minded, set-in-ways	curious, self-exploratory
Independence	outgoing, social	loner, crave solitude
Perfectionism	Disorganized, messy	orderly, thorough
Tension	relaxed, cool	stressed, unsatisfied

Figure 2.3: Cattell's personality characteristics

The Big Five: Five-Factor Model

As a result of a thorough research on Cattell's and Eysenck's personality trait theories, the Big Five theory was formulated [59, 60, 61]. This model states that there are 5 core traits which collaborate in order to form a single personality. These include:

- **Extraversion** (outgoing/energetic vs. solitary/reserved): tendency to be active, sociable, person-oriented, talkative, optimistic, empathetic.
- **Openness to Experience** (inventive/curious vs. consistent/cautious): tendency to be imaginative, curious, creative and may have unconventional beliefs and values.
- **Agreeableness** (friendly/compassionate vs. challenging/detached): tendency to be good-natured, kind-hearted, helpful, altruistic and trusting.
- **Conscientiousness** (efficient/organized vs. easy-going/careless): tendency to be hardworking, reliable, ambitious, punctual and self-directed.
- **Neuroticism** (sensitive/nervous vs. secure/confident): tendency to become emotionally unstable and may even develop psychological distress. This personality trait is also referred as *Emotional Stability*.

The Big Five Model (also called OCEAN model) is the most widely accepted model of personality in psychology today.

2.2.2 Suitability of Big Five Model

According to Digman [60], a lot of psychological studies independently came to the conclusion that five traits are sufficient in describing the human personality. "If a large number of rating scales is used and if the scope of the scales is very broad, the domain of personality descriptors is almost completely accounted for by five robust factors", as sustained by [1]. However, there is no full agreement on the meaning of each trait, since they are considered vague. For example there is some disagreement as far as the openness factor is concerned. Some prefer the notation "intellect" instead of "Openness to experience".

Moreover, the same model is utilized in different languages and cultures, such as Chinese [62] and Indian [63]. According to researchers, such as [64, 65], the Openness trait is particularly unsupported in asian cultures such as Chinese and Japanese. In this case a different fifth factor is sometimes identified in order to substitute the one used by the English culture. Also the relationship between language and personality has been investigated (see [66] for a survey), although most research is focus on the English language.

Despite all the problems and criticisms, the Big Five model is considered suitable for describing the individual's personality. This model has been established, since it is convenient for computational and learning approached and, moreover, general enough in order to be applied to many languages and cultural with the exception of the Openness trait that cannot be directly used in eastern civilizations.

Although the Big Five model sufficiently describes an affect processing system that outlines persistent human behavioral responses to broad classes of environmental stimuli, characterizing in this way a unique individual [67], personality nevertheless changes over time and adapts to the environment. For example, as DeYoung [68] pointed out, goals, motivations and context influence the way people display their personality. People may also pretend to have different personality traits depending on their intentions in the context of a specific situation. The general position of psychologists about these problems (that is also at the basis of Adelstein's [69] work) is that individuals have some rather fixed core personality traits and other more variable peripheral traits. However, such a fission has not been applied to the Big Five model and hence, the traits are influenced by external factors, such as the context of a given situation.

2.3 Machine Learning for Natural Language Processing

2.3.1 Natural Language Processing

Natural Language Processing (NLP) is a field of computer science, artificial intelligence (also called machine learning), and linguistics concerned with the interactions between computers and human (natural) languages. It is the process of a computer extracting meaningful information from natural language input and/or producing natural language output. It is analysis of human language based on semantics and various parsing techniques [70]. The goal of NLP is to identify the computational machinery needed for an agent to exhibit various forms of linguistic behavior (i.e. Scientific Goal). It also design, implement, and test systems that process natural languages for practical applications (i.e. Engineering Goal).

NLP is a discipline between linguistics and computer science which is concerned with the computational aspects of the human language faculty. The main task of it is to construct programs in order to process words and texts in natural language. The main aspects of NLP are:

- Information Retrieval (IR): It is concerned with storing, searching and retrieving of information from text documents. It is a field within computer science closer to databases and relies on some of the NLP methods.

- Machine Translation (MT): It is related to automatic translation from one human language to another [71].

- Language Analysis: It is concerned with parsing of an input sentence to construct syntactic tree and further sentiment analysis is done to find meaningful words in a sentence.

Linguistic is the science of language. It study includes Sounds (phonology), Word formation (morphology), Sentence structure (syntax), Meaning (semantics) and Understanding (pragmatics).

- Phonological Analysis: relates sounds to the words we recognize. Phoneme is smallest unit of sound, and the phones are aggregated into word sounds.

- Morphological Analysis: Morphology is a sub discipline of linguistics that studies word structure. It is concerned with derivation of new words from existing ones. In NLP, words are known as lexicon items and a set of words form a lexicon. Lexicon is a module that tells what words there are and what properties they have.
- Syntactic Analysis: is analysis of words in a sentence to know the grammatical structure of a sentence and these words are transformed into structures that show how the words relate to each others.
- Semantic Analysis: It is concerned with the meaning of the language. The first step in semantic processing system is to look up the individual words in a dictionary (or lexicon) and extract their meanings [72].
- Pragmatic Analysis: to reinterpret what was said to what was actually meant. It concerns how sentences are used in different situations and how use affects the interpretation of the sentence.

Most NLP applications such as information extraction, machine translation, sentiment analysis and question answering, require both syntactic and semantic analysis at various levels.

- Information Retrieval (IR) & Web Search: is the science of searching for documents, for information within documents, and for metadata about documents, as well as that of searching databases and the World Wide Web.
- Information Extraction (IE): is a type of information retrieval whose goal is to automatically extract structured information, i.e. categorized and contextually and semantically well-defined data from a certain domain, from unstructured machine-readable documents.
- Question Answering (QA): is the response from documents to extracted or generated answer.
- Text Summarization: Text Summarization is the process of distilling the most important information from a source in order to produce an abridged version.
- Machine Translation (MT): is the use of computer software in order to translate text or speech from one natural language to another.
- Speech Recognition & Synthesis: is the extraction of textual representation of a spoken utterance
- Natural Language Understanding and Generation (NLU, NLG): NLG system is like a translator that converts a computer based representation into a natural language representation.
- Human-Computer Conversation: is the dialogue between human and computer using natural language.
- Text Generation: A method for generating sentences from ”keywords” or ”headwords”.
- Hand writing recognition: Ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices.

2.3.2 Definition of Machine Learning

Machine Learning (ML) is a subfield of Artificial Intelligence (AI). The goal of ML generally is to understand the structure of data and fit that data into models that can be understood and utilized by people. Although ML is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. ML algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range.

Because of this, ML facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

In ML, tasks are generally classified into broad categories. These categories are based on how learning is received or how feedback on the learning is given to the system developed. Two of the most widely adopted ML methods are **supervised learning** which trains algorithms based on example input and output data that is labeled by humans, and **unsupervised learning** which provides the algorithm with no labeled data in order to allow it to find structure within its input data.

2.3.3 Definition of different ML methods

Supervised Learning

In supervised learning, there are input variables (x) and an output variable (Y) and the goal is learning the mapping function from the input to the output via an algorithm.

$$Y = f(X) \tag{2.1}$$

The goal is to approximate the mapping function so well that when new input data (x) are fed to the model, the corresponding output variables (Y) could be successfully predicted.

It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

Supervised learning problems can be further grouped into **regression** and **classification** problems.

- Regression is the problem of estimating or predicting a continuous quantity. What will be the value of the S&P 500 one month from today? How tall will a child be as an adult? How many of our customers will leave for a competitor this year? These are examples of questions that would fall under the umbrella of regression. To solve these problems in a supervised Machine Learning framework, we would gather past examples "right answer" input/output pairs that deal with the same problem. For the inputs, we would identify features that we believe would be predictive of the outcomes that we wish to predict.

- Classification deals with assigning observations into discrete categories, rather than estimating continuous quantities. In the simplest case, there are two possible categories; this case is known as binary classification. Many important questions can be framed in terms of binary classification. Will a given customer leave us for a competitor? Does a given patient have cancer? Does a given image contain a hot dog? Algorithms for performing binary classification are particularly important because many of the algorithms for performing the more general kind of classification where there are arbitrary labels are simply a bunch of binary classifiers working together.

Unsupervised Learning

On the other hand, there is an entirely different class of tasks referred to as unsupervised learning. Supervised learning tasks find patterns where we have a dataset of "right answers" to learn from. Unsupervised learning tasks find patterns where we don't. This may be because the "right answers" are unobservable, or infeasible to obtain, or maybe for a given problem, there isn't one "right answer".

A large subclass of unsupervised tasks is the problem of clustering. Clustering refers to grouping observations together in such a way that members of a common group are similar to each other, and different from members of other groups. A common application here is in marketing, where we wish to identify segments of customers or prospects with similar preferences or buying habits. A major

challenge in clustering is that it is often difficult or impossible to know how many clusters should exist, or how the clusters should look.

A very interesting class of unsupervised tasks is generative modelling. Generative models are models that imitate the process that generates the training data. A good generative model would be able to generate new data that resembles the training data in some sense. This type of learning is unsupervised because the process that generates the data is not directly observable–only the data itself is observable.

Reinforcement Learning

A newer type of learning problem that has gained a great deal of traction recently is called reinforcement learning. In reinforcement learning, we do not provide the machine with examples of correct input-output pairs, but we do provide a method for the machine to quantify its performance in the form of a reward signal. Reinforcement learning methods resemble how humans and animals learn: the machine tries a bunch of different things and is rewarded when it does something well.

Reinforcement learning is useful in cases where the solution space is enormous or infinite, and typically applies in cases where the machine can be thought of as an agent interacting with its environment. One of the first big success stories for this type of model was by a small team that trained a reinforcement learning model to play Atari video games using only the pixel output from the game as input [73]. The model was eventually able to outperform human players at three of the games.

2.3.4 Feature Engineering

Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. Feature engineering includes several steps as presented next.

Pre-processing

Pre-processing analyzes the opinions from syntactical point of view and original syntax of sentence is not disturbed. The several techniques like Part-Of-Speech (POS) tagging, stemming and stop word removal are applied to a data set for noise reduction and facilitating feature extraction.

POS tagging. POS tagging is a linguistic technique used since 1960 [74, 75, 76, 77, 78, 79, 80, 81]. POS tagging [82] assigns a tag to each word in a text and classifies a word to a specific morphological category such as norm, verb, adjective, etc. POS taggers are efficient for explicit feature extraction in terms of accuracy they achieved [74, 77, 79, 83]. Hidden Markov Models are widely used for developing POS taggers due to accuracy as compared to other techniques like rule based, statistical and machine learning [84]. Different English language POS taggers like NL Processor linguistic parser, Stanford POS tagger, Gate ANNIE POS Tagger and Claws POS tagger are used for this purpose. Python based NLTK toolkit [85] has a rich collection of all modules including POS, needed by NLP researchers and text miners.

Stemming and Lemmatization. Stemming and Lemmatization are two essential morphological processes of preprocessing module during feature extraction [74, 76, 77, 78, 79, 81, 86, 81]. The stemming process converts all the inflected words present in the text into a root form called a stem [82]. For example, ‘automatic’, ‘automate’ and ‘automation’ are each converted into the stem ‘automat.’ Stemming gives faster performance in applications where accuracy is not major issue. The first stemmer was published by Julie Beth Lovins in 1968. Martin Porter designed and published his stemmer in the July 1980. Porter and Lancaster are the stemming algorithms, supported by python NLTK. RSLP Stemmer1, ISRI Stemmer2 and SnowballStemmer3 are non-English plugins. The lemma of a word includes its base form plus inflected forms [87]. For example the words “plays”, “played and “playing” have “play” as their lemma. Lemmatization groups together various inflected forms of word into a single one. Stemming removes word inflections only whereas lemmatization replaces words with their base form. For example, the words “caring” and “cars” are reduced to “car” in a stemming

process whereas lemmatization reduces it to “care” and “car” respectively, hence lemmatization is considered to be more accurate. Unlike stemming, lemmatization needs additional dictionary support for searching and indexing, which enhances its accuracy in feature extraction applications, but degrades speed of lemmatizer. Word Net Lemmatizer with Word Net Database is used to lookup for lemmas.

Stop Word Removal. Stop word concept was first introduced by Hans Luhn, H.P [88]. Stop words are common and high frequency words like “a”, “the”, “of”, “and”, “an”. Different methods available for stop-word elimination [82]; ultimately enhance performance of feature extraction algorithm [74, 82, 77]. The stop words removal reduces dimensionality of the data sets and thus key words left in the corpus can be identified more easily by the automatic feature extraction techniques. Words to be removed are taken from a commonly available list of stop words. Savoy [89] had given huge collection stop word list. At simplest level stop words are iterated in chosen word list and removed from text.

Feature Selection

In the past thirty years, the dimensionality of the data involved in ML and data mining tasks has increased explosively. Data with extremely high dimensionality has presented serious challenges to existing learning methods [90], i.e., the curse of dimensionality [91]. With the presence of a large number of features, a learning model tends to overfit, resulting in their performance degenerates. To address the problem of the curse of dimensionality, dimensionality reduction techniques have been studied, which is an important branch in the ML and data mining research area. Feature selection is a widely employed technique for reducing dimensionality among practitioners. It aims to choose a small subset of the relevant features from the original ones according to certain relevance evaluation criterion, which usually leads to better learning performance (e.g., higher learning accuracy for classification), lower computational cost, and better model interpretability.

According to whether the training set is labeled or not, feature selection algorithms can be categorized into supervised [92, 93], unsupervised [94, 95] and semi-supervised feature selection [96, 97]. Supervised feature selection methods can further be broadly categorized into filter models, wrapper models and embedded models. The filter model separates feature selection from classifier learning so that the bias of a learning algorithm does not interact with the bias of a feature selection algorithm. It relies on measures of the general characteristics of the training data such as distance, consistency, dependency, information, and correlation. Relief [98], Fisher score [99] and Information Gain based methods [100] are among the most representative algorithms of the filter model. The wrapper model uses the predictive accuracy of a predetermined learning algorithm to determine the quality of selected features. These methods are prohibitively expensive to run for data with a large number of features.

Due to these shortcomings in each model, the embedded model, was proposed to bridge the gap between the filter and wrapper models. First, it incorporates the statistical criteria, as filter model does, to select several candidate features subsets with a given cardinality. Second, it chooses the subset with the highest classification accuracy [101]. Thus, the embedded model usually achieves both comparable accuracy to the wrapper and comparable efficiency to the filter model. The embedded model performs feature selection in the learning time. In other words, it achieves model fitting and feature selection simultaneously [102, 103].

Many researchers also paid attention to developing unsupervised feature selection. Unsupervised feature selection is a less constrained search problem without class labels, depending on clustering quality measures, and can eventuate many equally valid feature subsets. With high-dimensional data, it is unlikely to recover the relevant features without considering additional constraints. Another key difficulty is how to objectively measure the results of feature selection [104]. A comprehensive review about unsupervised feature selection can be found in [105].

Supervised feature selection assesses the relevance of features guided by the label information but a good selector needs enough labeled data, which is time consuming. While unsupervised feature selection works with unlabeled data but it is difficult to evaluate the relevance of features. It is common

to have a data set with huge dimensionality but small labeled-sample size. High-dimensional data with small labeled samples permits too large a hypothesis space yet with too few constraints (labeled instances). The combination of the two data characteristics manifests a new research challenge. Under the assumption that labeled and unlabeled data are sampled from the same population generated by target concept, semi-supervised feature selection makes use of both labeled and unlabeled data to estimate feature relevance [97].

Feature weighting is thought of as a generalization of feature selection. In feature selection, a feature is assigned a binary weight, where 1 means the feature is selected and 0 otherwise. However, feature weighting assigns a value, usually in the interval $[0, 1]$ or $[-1, 1]$, to each feature. The greater this value is, the more salient the feature will be. Most of feature weight algorithms assign a unified (global) weight to each feature over all instances. However, the relative importance, relevance and noise in the different dimensions may vary significantly with data locality. There are local feature selection algorithms where the local selection of features is done specific to a test instance, which is is common in lazy leaning algorithms such as k-Nearest Neighbor (kNN) [106, 107]. The idea is that feature selection or weighting is done at classification time (rather than at training time), because knowledge of the test instance sharpens the ability to select features.

Typically, a feature selection method consists of four basic steps, namely, subset generation, subset evaluation, stopping criterion, and result validation. In the first step, a candidate feature subset will be chosen based on a given search strategy, which is sent, in the second step, to be evaluated according to certain evaluation criterion. The subset that best fits the evaluation criterion will be chosen from all the candidates that have been evaluated after the stopping criterion are met. In the final step, the chosen subset will be validated using domain knowledge or a validation set.

2.3.5 Traditional ML Approaches

Naive Bayes

Naive Bayes is a popular and straightforward algorithm for binary and multi-class classification tasks. It was firstly applied for filtering junk e-mails on the Internet [108]. It is called Naive Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. Also, as derived from the name, it is based on the Bayes' Theorem, according to which:

$$P(h|d) = \frac{P(d|h)P(h)}{P(d)} \tag{2.2}$$

- $P(h|d)$: probability of hypothesis h given the data d. This is called the posterior probability.
- $P(d|h)$: probability of data d given that the hypothesis h was true.
- $P(h)$: probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.
- $P(d)$: probability of the data (regardless of the hypothesis).

When using the Naive Bayes classifier we are interested in calculating the posterior probability of $P(h|d)$ from the prior probability $P(h)$ with $P(D)$ and $P(d|h)$. After calculating the posterior probability for a number of different hypotheses, you can select the hypothesis with the highest probability. This is the maximum probable hypothesis and may formally be called the Maximum A Posteriori (MAP) hypothesis. This can be written as:

$$MAP(h) = max(P(h|d)) = max(\frac{P(d|h)P(h)}{P(d)}) = max(P(d|h)P(h)) \tag{2.3}$$

The $P(d)$ is a normalizing term which allows us to calculate the probability. We can drop it when we are interested in the most probable hypothesis as it is constant and only used to normalize. Focusing on the classification task, if we have an even number of instances in each class in our training data, then the probability of each class (e.g. $P(h)$) will be equal.

Support Vector Machines

Support Vector Machines (SVMs) are perhaps one of the most popular and talked about machine learning algorithms. They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high-performing algorithm with little tuning.

SVMs are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. A schematic example is shown in Figure 2.4. In this example, the objects belong either to class ”blue” or ”red”. The separating line defines a boundary on the right side of which all objects are ”blue” and to the left of which all objects are ”red”. Any new object (white circle) falling to the right is labeled, i.e., classified, as ”blue” (or classified as ”red” should it fall to the left of the separating line).

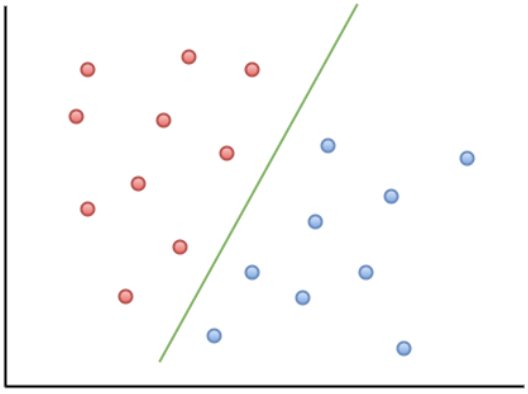


Figure 2.4: Linear classification using SVM

The above is a classic example of a linear classifier, i.e., a classifier that separates a set of objects into their respective groups (”blue” and ”red” in this case) with a line. Most classification tasks, however, are not that simple, and often more complex structures are needed in order to make an optimal separation, i.e., correctly classify new objects (test cases) on the basis of the examples that are available (train cases). Classification tasks that require drawing separating lines to distinguish between objects of different class memberships are known as hyperplane classifiers. SVMs are particularly suited to handle such tasks. A SVM model maps the original objects (left side of the schematic) into a rearrangement, using a set of mathematical functions, known as kernels. The process of rearranging the objects is known as transformation.

Random Forest

The random forest algorithm is an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests do not tend to overfitting in comparison to decision trees.

Decision tree concept is more to the rule based system. Given the training dataset with targets and features, the decision tree algorithm will come up with some set of rules. The same set rules can be used to perform the prediction on the test dataset. The pseudocode used in decision trees can be summarized as follows:

1. Place the best attribute of the dataset at the root of the tree.

2. Split the training set into subsets. Subsets should be made in such a way that each subset contains data with the same value for an attribute.

3. Repeat step 1 and step 2 on each subset until you find leaf nodes in all the branches of the tree.

In decision trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node. We continue comparing our record's attribute values with other internal nodes of the tree until we reach a leaf node with predicted class value.

A random forest classifier has all the hyperparameters of a decision-tree classifier and also all the hyperparameters of a bagging classifier, to control the ensemble. The random forest algorithm brings extra randomness into the model, when it is growing the trees. Instead of searching for the best feature while splitting a node, it searches for the best feature among a random subset of features. This process creates a wide diversity, which generally results in a better model. Therefore when you are growing a tree in random forest, only a random subset of the features is considered for splitting a node. You can even make trees more random, by using random thresholds on top of it, for each feature rather than searching for the best possible thresholds (like a normal decision tree does). The psedocode for the random forest algorithm is presented below:

1. Randomly select k features from total m features, where $k << m$

2. Among the k features, calculate the node d using the best split point.

3. Split the node into daughter nodes using the best split.

4. Repeat 1 to 3 steps until l number of nodes has been reached.

5. Build forest by repeating steps 1 to 4 for n number times to create n number of trees.

The procedure of training the random forest classifier is also illustrated in Figure 2.5[1].

After training the classifier as described above, the prediction procedure follows:

1. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)

2. Calculate the votes for each predicted target.

3. Consider the high voted predicted target as the final prediction from the random forest algorithm.

2.3.6 Word Embeddings

Word embeddings are commonly used in many NLP tasks because they are found to be useful representations of words and often lead to better performance in the various tasks performed. Word embeddings allow words to be represented by a series of numbers.

Word2vec

Word2vec [109] is a two-layer neural net that processes text. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus. While Word2vec is not a deep neural network (see Section 2.4), it turns text into a numerical form that deep nets can understand.

The purpose and usefulness of Word2vec is to group the vectors of similar words together in vectorspace. That is, it detects similarities mathematically. Word2vec creates vectors that are distributed numerical representations of word features, features such as the context of individual words. It does so without human intervention.

[1] http://dataaspirant.com

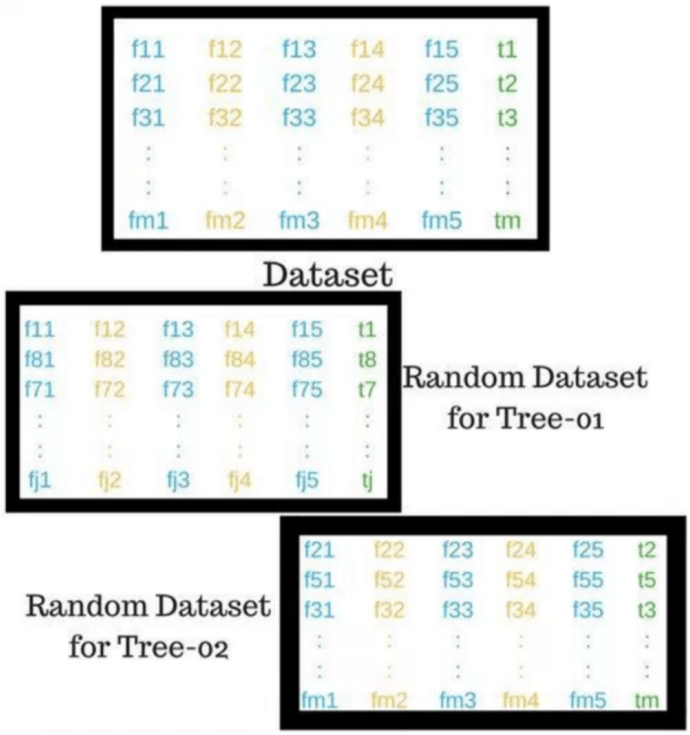


Figure 2.5: Procedure of training the random forest classifier

Given enough data, usage and contexts, Word2vec can make highly accurate guesses about a word's meaning based on past appearances. Those guesses can be used to establish a word's association with other words (e.g. "man" is to "boy" what "woman" is to "girl"), or cluster documents and classify them by topic. Those clusters can form the basis of search, sentiment analysis and recommendations in such diverse fields as scientific research, legal discovery, e-commerce and customer relationship management.

The output of the Word2vec neural net is a vocabulary in which each item has a vector attached to it, which can be fed into a deep learning net or simply queried to detect relationships between words.

Measuring cosine similarity, no similarity is expressed as a 90 degree angle, while total similarity of 1 is a 0 degree angle, complete overlap.

The vectors we use to represent words are called neural word embeddings, and representations are strange. Word2vec is similar to an autoencoder, encoding each word in a vector, but rather than training against the input words through reconstruction, as a restricted Boltzmann machine does, word2vec trains words against other words that neighbor them in the input corpus. It does so in one of two ways, either using context to predict a target word (a method known as continuous bag of words, or **CBOW**), or using a word to predict a target context, which is called **skip-gram**. These two methods are presented in Figure 2.6.

Being probabilistic in nature, these are supposed to perform superior to deterministic methods (generally). Though CBOW (predict target from context) and skip-gram (predict context words from target) are just inverted methods to each other, they each have their advantages/disadvantage. Since CBOW can use many context words to predict the 1 target word, it can essentially smooth out over the distribution. This is essentially like regularization and is offer very good performance when our input data is not so large. However the skip-gram model is more fine grained so we are able to extract more information and essentially have more accurate embeddings when we have a large data set (large data is always the best regularizer). Skip-gram with negative sub-sampling outperforms every other method generally.

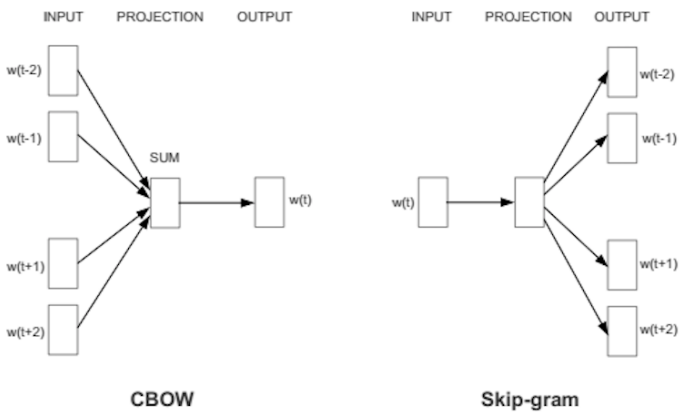


Figure 2.6: CBOW and skip-gram methods for training word2vec embeddings

GloVe

In contrast to word2vec, GloVe [110] seeks to make explicit what word2vec does implicitly: Encoding meaning as vector offsets in an embedding space — seemingly only a serendipitous by-product of word2vec — is the specified goal of GloVe.

Specifically, GloVe illustrate that the ratio of the co-occurrence probabilities of two words (rather than their co-occurrence probabilities themselves) is what contains information and so look to encode this information as vector differences.

For this to be accomplished, GloVe method directly aims to reduce the difference between the dot product of the vectors of two words and the logarithm of their number of co-occurrences. As co-occurrence counts can be directly encoded in a word-context co-occurrence matrix, GloVe takes such a matrix rather than the entire corpus as input.

2.4 Deep Neural Networks

2.4.1 Introduction

In the past 10 years, the best-performing artificial-intelligence systems — such as the speech recognizers on smartphones or Google's latest automatic translator — have resulted from a technique called "deep learning." Deep learning is in fact a new name for an approach to artificial intelligence called neural networks, which have been going in and out of fashion for more than 70 years. Neural networks were first proposed in 1944 by Warren McCullough and Walter Pitts, two University of Chicago researchers who moved to MIT in 1952 as founding members of what's sometimes called the first cognitive science department. Neural nets were a major area of research in both neuroscience and computer science until 1969, when, according to computer science lore, they were killed off by the MIT mathematicians Marvin Minsky and Seymour Papert, who a year later would become co-directors of the new MIT Artificial Intelligence Laboratory. The technique then enjoyed a resurgence in the 1980s, fell into eclipse again in the first decade of the new century, and has returned like gangbusters in the second, fueled largely by the increased processing power of graphics chips.

2.4.2 Artificial Neural Networks

An Artificial Neuron Network (ANN), popularly known as Neural Network is a computational model based on the structure and functions of biological neural networks. It is inspired by the human brain and aims at receiving, processing, and transmitting information in terms of Computer Science.

The exact workings of the human brain are still a mystery. Yet, some aspects of this amazing processor are known. In particular, the most basic element of the human brain is a specific type of cell

which, unlike the rest of the body, doesn't appear to regenerate. Because this type of cell is the only part of the body that isn't slowly replaced, it is assumed that these cells are what provides us with our abilities to remember, think, and apply previous experiences to our every action. These cells, all 100 billion of them, are known as neurons. Each of these neurons can connect with up to 200,000 other neurons, although 1,000 to 10,000 is typical. The power of the human mind comes from the sheer numbers of these basic components and the multiple connections between them. It also comes from genetic programming and learning.

The individual neurons are complicated. An example of the structure of a biological neuron is depicted in Figure 2.7[2] They have a myriad of parts, sub-systems, and control mechanisms. They convey information via a host of electrochemical pathways. There are over one hundred different classes of neurons, depending on the classification method used. Together these neurons and their connections form a process which is not binary, not stable, and not synchronous. In short, it is nothing like the currently available electronic computers, or even ANNs.

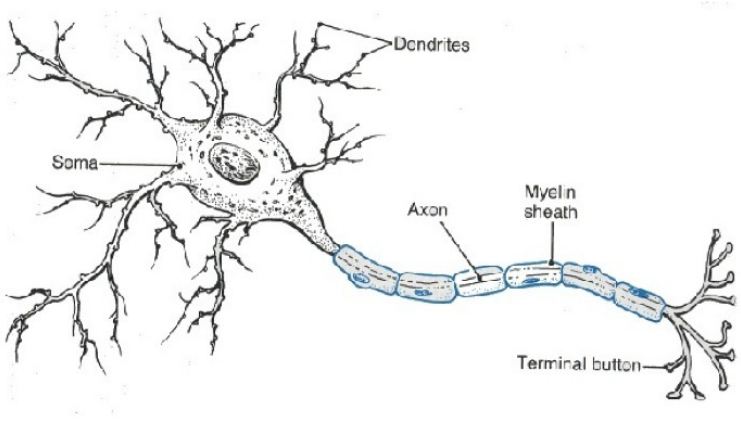


Figure 2.7: Overview of a biologican neuron

These ANNs try to replicate only the most basic elements of this complicated, versatile, and powerful organism. Specifically, the goal of ANNs is not the grandiose recreation of the brain. On the contrary, neural network researchers are seeking an understanding of nature's capabilities for which people can engineer solutions to problems that have not been solved by traditional computing.To do this, the basic unit of neural networks, the artificial neurons, simulate the four basic functions of natural neurons. Figure 2.8 shows a fundamental representation of an artificial neuron.

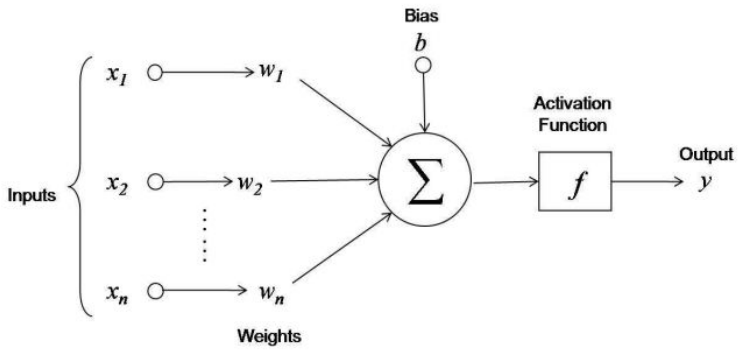


Figure 2.8: Overview of the functions of an artificial neuron

As presented in Figure 2.8, various inputs to the network are represented by the mathematical symbol, x_n. Each of these inputs are multiplied by a connection weight. These weights are represented

[2] http://www.neuropsychologysketches.com/Neurons.html

by w_n. In the simplest case, these products are simply summed, fed through a transfer function f to generate a result, and then produce output y. In the more general case, there are various transfer (activation) functions in terms of an artificial neuron.

In order to learn complex non-linear functions, architectures that combine several artificial neurons can be designed and implemented. Such architectures are called Multi-Layer Perceptrons (MLPs). Instead of MLPs, Feed-Forward Neural Networks (FFNNs) can also be implemented, where each neuron connects with all neurons of the previous layer and there are no connections between the neurons of the same layer.

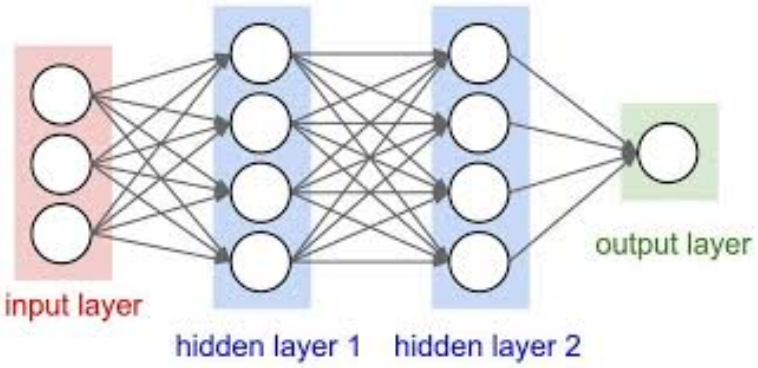


Figure 2.9: Overview of the architecture of an ANN

The structure of an ANN is depicted in Figure 2.9[3]. As observed the network consists of the following layers:

- Input layer: All the inputs are fed to the network via this layer.
- Hidden layer(s): Through this layer the input is processed and the features are extracted by the network. When moving to higher hidden layers, high-level features are constructed.
- Output layer: After the data processing, a decision is made by the network in this layer.

If a neural network consists of more than one hidden layers then it is referred as ”Deep Neural Network” (DNN). When increasing the depth of a neural network and uilizing non-linear transfer functions, more complex functions can be learnt.

Training

A key feature of the training procedure of neural networks is an iterative learning process in which data cases are presented to the network one at a time, and the weights associated with the input values are adjusted each time. After all cases are presented, the process often starts over again. During this learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of input samples. In this section the main procedures required for training is going to be presented and described.

Activation function. Activation functions are important for an ANN to learn and understand the complex patterns. The main function of it is to introduce non-linear properties into the network. What it does is, it calculates the ”weighted sum” and adds direction and decides whether to ”fire” a particular neuron or not. The non linear activation function helps the model to understand the complexity and give accurate results.

The functions that are commonly used as an activation function in an ANN are:

[3] https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/

- *Sigmoid function*: The sigmoid function is an activation function where it scales the values between 0 and 1 by applying a threshold and can be described by the following equation:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.4}$$

The illustration of this function is presented in Figure 2.10. When the weighted sum is applied in the place of x, the values are scaled in between 0 and 1. The value never reaches zero nor exceed 1 in the above equation. The large negative numbers are scaled towards 0 and large positive numbers are scaled towards 1.

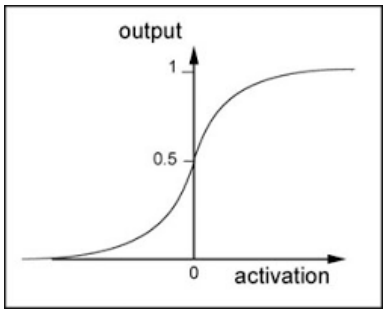


Figure 2.10: Sigmoid function

- *Hyperbolic tangent*: The tanh function is an activation function which re-scales the values between -1 and 1 by applying a threshold just like a sigmoid function. The advantage of this function is that the values of a tanh is zero centered which helps the next neuron during propagating. This function is described as follows:

$$f(x) = tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.5}$$

The illustration of this function is presented in Figure 2.11. When the weighted sum of the inputs is applied in the tanh(x), it re-scales the values between -1 and 1. The large negative numbers are scaled towards -1 and large positive numbers are scaled towards 1.

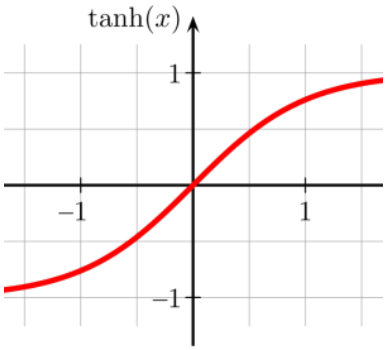


Figure 2.11: Hyperbolic tangent function

- *Rectified Linear Unit (ReLU)*: This is one of the most widely used activation functions. The benefits of ReLU is the sparsity, it allows only values which are positive and negative values are not passed which will speed up the process and it will negate or bring down possibility of occurrence of a dead neuron. The mathematic description of this function is presented below:

$$f(x) = (0, max) \tag{2.6}$$

This function will allow only the maximum values to pass during the front propagation as shown in Figure 2.12. The draw backs of ReLU is when the gradient hits zero for the negative values, it does not converge towards the minima which will result in a dead neuron while back propagation.

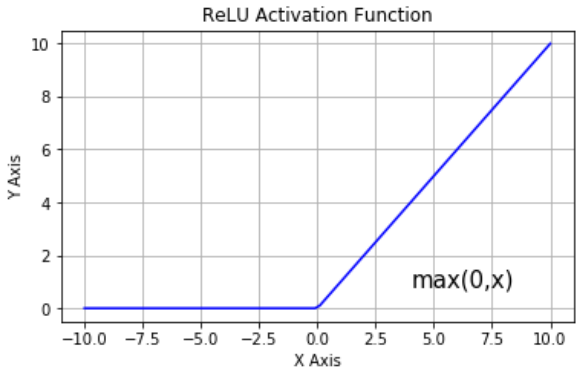


Figure 2.12: ReLU function

- *Leaky ReLU*: This function overcomes the disadvantage of ReLU by allowing a small negative value during the back propagation in case of a dead ReLU problem. This will eventually activate the neuron and bring it down. This function is described by:

$$f(x) = 1(x < 0)(\alpha x) + 1(x >= 0)(x) \tag{2.7}$$

where α is a small constant

The illustration of this function is presented in Figure 2.13. This activation function also has drawbacks, during the front propagation if the learning rate is set very high it will overshoot killing the neuron. This will happen when the learning rate is not set at an optimum level.

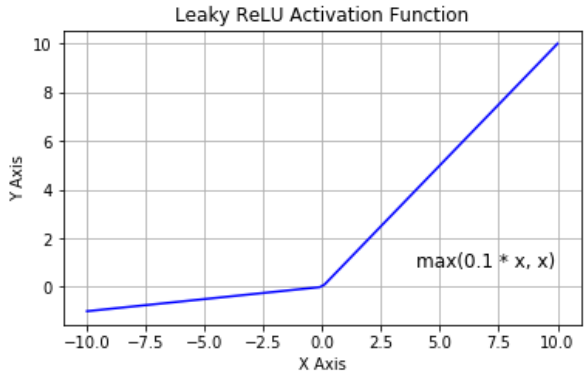


Figure 2.13: Leaky ReLU function

Cost function. The cost function evaluates the performance of the model. Specifically, it computes how well a set of parameters W can estimate the real values on a data set. The kind of used cost function depends on the nature of the problem to be solved.

In ANNs, the *softmax* activation function is typically applied to the output of the last layer. The softmax function is described by the following equation:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{2.8}$$

This function compress the values of a K-dimensional vector z in the space between 0 and 1 aiming at summing all values at 1. By applying this function to the values of a vector, the values of all classes of a problem are regularized in order to respond to respective probabilities.

The most commonly used cost function is the cross-entropy loss, presented in Equation 2.9. This function is used in ANNs as described in Equation 2.10

$$L_i = -log(\frac{e^{y_i}}{\sum_j e^{f_j}}) \tag{2.9}$$

$$L(W) = -\frac{1}{n}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij} log(p_{ij}) \tag{2.10}$$

where W are the weights of the network that need estimation, i are the observations fed to the network, N is the total number of observations contained in the dataset, j are the classes of the classification problem, K is the total number of the classes and y is the estimated value for the i_th observation.

This function computes the distance between two probability distributions. When using an ANN in order to solve a classification problem the output of the network for a single observation is a distribution of probabilities over all the classes of the problem. This distribution is then compared with the one-hot vector that represents the true class of the observation.

Regularization. Regularization is a key component in preventing overfitting. Also, some techniques of regularization can be used to reduce model capacity while maintaining accuracy, for example, to drive some of the parameters to zero. This might be desirable for reducing model size or driving down cost of evaluation in mobile environment where processor power is constrained.

The regularization methods commonly used are:

- *Early-stopping*: It combats overfitting interrupting the training procedure once model's performance on a validation set gets worse. A validation set is a set of examples that we never use for gradient descent, but which is also not a part of the test set. The validation examples are considered to be representative of future test examples. Early stopping is effectively tuning the hyper-parameter number of epochs/steps. Intuitively as the model sees more data and learns patterns and correlations, both training and test error go down. After enough passes over training data the model might start overfitting and learning noise in the given training set. In this case training error would continue going down while test error (how well we generalize) would get worse. Early stopping is all about finding this right moment with minimum test error.

- *Dropout* [111] : At each training iteration a dropout layer randomly removes some nodes in the network along with all of their incoming and outgoing connections. Dropout can be applied to hidden or input layer. The dropout is considered as a good regularization method for the following reasons. The nodes become more insensitive to the weights of the other nodes (co-adaptive), and therefore the model is more robust. If a hidden unit has to be working well with different combinations of other hidden units, it's more likely to do something individually useful. Moreover, dropout can be viewed as a form of averaging multiple models ("ensemble"), technique which shows better performance in most machine learning tasks (e.g. ensemble training is the intuition behind random forests or gradient boosting decision trees).

- *Weight penalty*: It is standard way for regularization. It relies strongly on the implicit assumption that a model with small weights is somehow simpler than a network with large weights. The penalties try to keep the weights small or non-existent (zero) unless there are big gradients to counteract it, which makes models also more interpretable. An alternative name in literature for weight penalties is *weight decay* since it forces the weights to decay towards zero.

– *L2 norm*: L2 norm penalizes the square value of the weight. This normalization technique tends to drive all the weights to smaller values.

$$L2 = \sum_{i=0}^{N} w_i^2 \tag{2.11}$$

– *L1 norm*: penalizes the absolute value of the weight (it is a v-shape function). This technique tends to drive some weights to exactly zero (introducing sparsity in the model), while allowing some weights to be big.

$$L1 = \sum_{i=0}^{N} |w_i| \tag{2.12}$$

Optimization. Optimization algorithms helps us to minimize (or maximize) a cost function. Gradient Descent (GD) is one of the most popular algorithms to perform optimization and by far the most common way to optimize neural networks. GD is a way to minimize an objective function $J(\theta)$. $J(\theta)$ is parameterized by a model's parameters by updating the parameters in the opposite direction of the gradient of the objective function $\nabla_\theta J(\theta)$ with respect to the parameters. The learning rate η determines the size of the steps we take to reach a (local) minimum. In other words, we follow the direction of the slope of the surface created by the objective function downhill until we reach a valley. The simple GD algorithm computes the gradient of the cost function with respect to the parameters θ for the entire training dataset:

$$\theta = \theta - \nabla_\theta J(\theta) \tag{2.13}$$

In contrast, the Stochastic Gradient Descent (SGD) performs a parameter update for each training example x_i and label y_i:

$$\theta = \theta - \nabla_\theta J(\theta; x_i; y_i) \tag{2.14}$$

GD performs redundant computations for large datasets, as it recomputes gradients for similar examples before each parameter update. SGD does away with this redundancy by performing one update at a time. It is therefore usually much faster and can also be used to learn online. Except for SGD there are also other alterations for optimization such as the Nesterov Momentum method [112]. Currently, optimization methods with an automatic regulation of the learning rate are used such as Adagrad [113], Adadelta [114] and Adam [115], which is the most widely used optimization technique.

Backpropagation. An ANN can be represented by a directive graph, where each node corresponds to a specific weight of the network. In order to update the weights of the network after each computation of the cost function the backpropagation algorithm [116] is utilized. The key element of backpropagation is an expression for the partial derivative $\frac{\partial C}{\partial w}$ of the cost function C with respect to any weight w (or bias b) in the network. The expression indicates how quickly the cost changes when the weights and biases are changed. The advantages of the backpropagation are not restricted to the fast computation, but also it provides an intuitive interpretation of how changing the weights and biases changes the overall behaviour of the network.

2.4.3 Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a class of ANN where connections between units form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Unlike FFNNs, RNNs can use their internal state (memory) to process sequences of inputs. In order to achieve it, the RNN creates the networks with loops in them, which allows it to persist the information. This loop structure allows the neural network to take the sequence of input, as presented in Figure 2.14[4].

[4] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

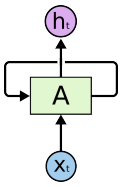


Figure 2.14: Loops in RNNs

The function of a RNN can be better understood in Figure 2.15 First, it takes the x_0 from the sequence of input and then it outputs h_0 (hidden state of the RNN) which together with x_1 is the input for the next step. So, the h_0 and x_1 is the input for the next step. Similarly, h_1 from the next is the input with x_2 for the next step and so on. This way, it keeps remembering the context while training.

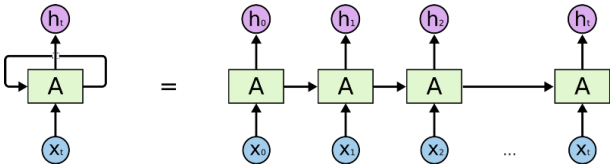


Figure 2.15: Unrolled version of a RNN

Hence, for each time step t the equations that describe the function of the RNN are:

$$h_t = f_h(W_h x_t + U_h h_{t-1} + b_h) \tag{2.15}$$

$$y_t = f_y(W_y h_t + b_y) \tag{2.16}$$

where

- h_t: the hidden state at time step t
- x_t: vector of the sequence input at time step t
- y_t: the output of the RNN at time step t
- W_x, U_h, W_y: the weights of the network for h, x, y, respectively
- b_h: bias for h
- f_x, f_h: activation functions for x and h, respectively

Bidirectional RNN

A Bidirectional RNN (BiRNN) consists of the combination of two different RNNs, where each one processes the input sequence with a different direction. The motivation in using BiRNNs is to produce a more accurate representation of the input sequence. For this purpose, there is a right-directed RNN which processes progressively the sequence by starting from the vector x_0 of the input and a left-directed RNN which starts the processing of the sequence from the vector x_T, where T is the length of the input sequence. Therefore, at each time step t, the equations that decribe a BiRNN are:

$$h_i = \vec{h_i} || \overleftarrow{h_i}, h_i \in \Re^{2N} \tag{2.17}$$

where $||$ stands for the concatenation of the two vectors and N is the dimensionality of each RNN.

Attention mechanism

Not all vectors that consist the input sequence contribute equally to the meaning that is expressed in the overall input. For this reason, an attention mechanism [117, 118] can be utilized in order to find the relative contribution (importance) of each input vector of a sequence. The attention mechanism assigns a weight a_i to each vector annotation h_i. The fixed representation r of the whole input is computed. as the weighted sum of all the word annotations.

$$e_i = tanh(W_h h_i + b_h), \quad e_i \in [-1, 1] \tag{2.18}$$

$$a_i = \frac{exp(e_i)}{\sum_{t=1}^{T} exp(e_t)}, \quad \sum_{i=1}^{T} a_i = 1 \tag{2.19}$$

$$r = \sum_{i=1}^{T} a_i h_i, \quad r \in R^{2L} \tag{2.20}$$

where W_h and b_h are the attention layer's weights.

Long Short-Term Memory

There are two major obstacles RNN's have or had to deal with:

- *Exploding Gradients*: The problem when the algorithm assigns very big weights to the network.
- *Vanishing Gradients*: The problem when the values of a gradient are too small and the model stops learning or takes too long because of that.

Long Short-Term Memory (LSTM) networks introduced by [119] can overcome these problems. LSTM is a variant of RNN that has the benefit of preserving long-distance dependencies between words and distilling unimportant words from the cell gate through its forget gate layer.

Specifically, in a RNN there are three gates: input i, forget f and output o gate. These gates determine whether or not to let new input in (input gate), delete the information because it is not important (forget gate) or to let it impact the output at the current time step (output gate). The gates in a LSTM are analog, in the form of sigmoids, meaning that they range from 0 to 1. The fact that they are analog, enables them to do backpropagation with it. The problematic issues of vanishing gradients is solved through LSTM because it keeps the gradients steep enough and therefore the training relatively short and the accuracy high. The structure of a LSTM cell is illustrated in Figure 2.16.

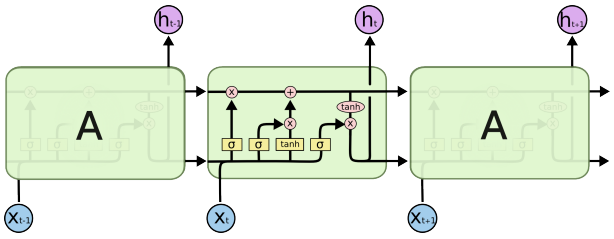


Figure 2.16: Stucture of a LSTM cell

In particular, given a sequence $x_1, x_2, ..., x_t, ..., x_l$ of vectors for an input sequence of length l, for the t^{th} vector x_t, with inputs h_{t-1} and c_{t-1}, h_t and c_t are computed as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \tag{2.21}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \tag{2.22}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{2.23}$$

$$u_t = tanh(W_u x_t + U_u h_{t-1} + b_u), \tag{2.24}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot u_t, \tag{2.25}$$

$$h_t = o_t \odot tanh(c_t), \tag{2.26}$$

where

- $W_j \in \Re^{d \times d}, U_j \in \Re^{d \times m}$ for $j \in \{i, f, o, u\}$ are weight matrices
- $b_j \in \Re^d$ are bias vectors
- $\sigma(\cdot)$ is the element-wise signoid function
- $tanh(\cdot)$ is the hyperbolic tangent function
- $\odot$ is the element-wise multiplication

2.4.4 Transfer Learning

Transfer learning (TL) has revolutionized Computer Vision (CV), but existing approaches in NLP still require task-specific modifications and training from scratch. TL plays a crucial role when a given dataset has insufficient labeled examples to train an accurate model. In such scenarios, the knowledge accumulated within a model pre-trained on a source dataset can be transferred to a target dataset, resulting in the improvement of the target model.

Specifically there are two different scenarios where TL can be applied successfully:

- Transferring knowledge to a semantically similar/same task but with a different dataset.
  - Source task (S) - A Large dataset for a specific task
  - Target task (T) - A small dataset for the same task
- Transferring knowledge to a task that is semantically different but shares the same neural network architecture so that neural parameters can be transferred.
  - Source task (S) - A large dataset for a specific task
  - Target task (T) - A small dataset for a semantically different task

Traditionally, only the first layer of any deep neural network is pretrained (see Section 2.3.6). Fine-tuning pretrained word embeddings [120], a simple TL technique that only targets a model's first layer has had a great impact in practice.

Recently, methods that go beyond transferring word embeddings were proposed. There are three different approaches used in terms of TL:

- **Pre-trained features**: The final fully-connected layers of deep neural networks are generally assumed to capture information that is relevant for solving the task (here on the source dataset). For a new task, we can thus simply use the off-the-shelf features extracted from the source task and concatenate them with the features of the final fully-connected layer for the target dataset. In this case, both the features from the source and the target dataset will contribute to the final prediction in the target dataset.
- **Parameter Initilization**: The Parameter Initilization (INIT) approach first trains the network on the source dataset, and then directly uses the tuned parameters to initialize the network for the target dataset. After transfer, we may fix the parameters in the target domain, i.e. fine tuning the parameters of the target dataset.
- **Multi-task learning**: MUlti-Task Learning (MULT), on the other hand, simultaneously trains samples in both domains.

Chapter 3

Dialogue Act Representation and Classification Using Recurrent Neural Networks

3.1 Introduction

Intent recognition in the context of dialogue is relatively close to the concept of DA recognition. DA can be interpreted as specific speech acts with a certain communicative function and semantic content [121]. The semantic content specifies the objects, propositions, events, etc. that the dialogue act is about; the communicative function specifies of the way an addressee should use the semantic content to update his information state.

The notion of "intent" is considered as a dimension of DAs. In other words, an intent (dimension) is a class of dialogue acts concerned with one particular aspect of communication that a dialogue act can address independently from other dimensions [122].

Dialogue Act (DA) classification constitutes a major processing step in Spoken Dialogue Systems (SDS) assisting the understanding of user input. Typically, this is implemented as the assignment of tags to user utterances that (lexically) describe the respective acts. DAs can be regarded as the minimal units of linguistic communication that are directly connected with the speaker's communicative intentions [123]. The output of DA classification can be exploited by other SDS components including the modules of natural language understanding and dialogue management.

Various approaches have been used for DA classification including Bayesian Networks (BN), Hidden Markov Models (HMM) [124], feed-forward Neural Networks [125], Decision Trees [43] and Support Vector Machines (SVM) [126]. The majority of these approaches examined both the utterance meaning as well as the sequence of the utterances within the dialogue. Recently, Deep Neural Networks (DNNs) have been utilized for dialogue act classification [127, 128, 129, 125] providing a significant increase in classification accuracy in task-independent conversations.

A challenge in the area of DA classification is the construction of models that are domain-agnostic and perform well across different granularities (coarse- vs. fine-grained) of DA tags. In recent deep learning approaches (e.g., [127, 129, 128]) DNNs rely on word embeddings that are generic or randomly set, ignoring domain-specific semantics. In [128], the performance of DA systems using various domain generic word embedding schemes was investigated and it was shown that performance depends on the granularity of DA tags.

We address the incorporation of DA-specific semantics in the framework of RNNs. Specifically, we propose a novel scheme for the automatic encoding of DA semantics via the extraction of a set of semantically salient keywords. Those keywords can be regarded as members of semantic subspaces that correspond to the respective DA. The importance of such keywords being relative to each DA is estimated by a regression model that exploits word embeddings. The classification of an unknown utterance relies on the computation of semantic similarity scores between the utterance words and the aforementioned DA subspaces, which are given as features in the used DNN in addition to typical word embeddings.

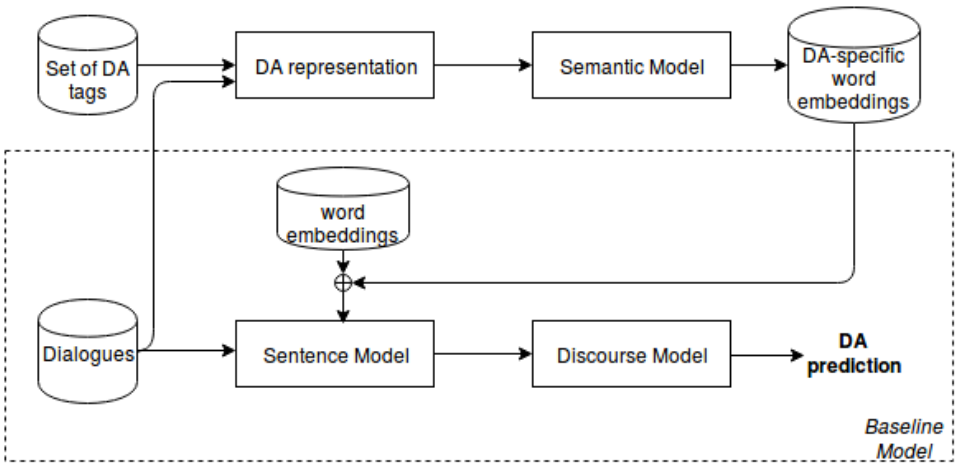


Figure 3.1: Overview of the proposed model.

3.2 Related Work

The early approaches of DA classification took advantage of lexical information, syntax, semantics, prosody, and dialogue history with manual extraction of the features [130, 124, 131, 132, 133, 134]. [130] built speech act classifiers in message board posts utilizing lexical, syntactic and semantic features by creating fixed, topic specific lexicons with keywords. [124] exploited lexical, collocational and prosodic cues, extracted from dialogues, in combination with discourse information of the DA sequence. The reported model is a Hidden Markov Model (HMM), where each HMM state corresponds to a sequential DA, achieving classification accuracy of 71.0% when applied to the Switchboard-DAMSL corpus [42]. [134] examined the role of affective analysis through affective lexicons in the recognition of DAs. In terms of affective text analysis, semantic features have been extracted based on the distributional semantic models built by [135].

Recently, the evolution of deep learning allowed the implementation of different models of DNNs in NLP, including the dialogue act classification. [127] used a mixture of Convolutional Neural Networks (CNNs) as a sentence model for the extraction of features from each utterance and Recurrent Neural Networks (RNNs) as a discourse model for the extraction of information about the sequence of the DA. This work improved the state-of-the-art DA classification on Switchboard-DAMSL corpus, reaching 73.9% accuracy. [128] built a model based on RNN and CNN that incorporates the preceding utterances via a two-layer feedforward Artificial Neural Network (ANN) for the extraction of discourse information. [125] proposed a hybrid architecture that combines an RNN sentence model with discourse information about the relation between two sequential utterances in the form of a latent variable. When the likelihood of the discourse relations derived from the model is maximized, treating the sentence model as a collateral factor in DA classification, an accuracy of 77.0% is achieved. [129] employed a deep Long Short Term Memory (LSTM) [119] structure with pre-trained word embeddings, and reported a classification accuracy of 80.1% outperforming the state-of-the-art.

For testing the various models suggested for DA classification accuracy, a variety of annotation schemes as well as datasets have been utilized [42, 43, 136, 137]. [42] provided a dataset annotated with 42 DA tags according to the Dialog Act Markup in Several Layers (DAMSL) [41] annotation scheme. [43] proposed an annotation scheme of five classes based on the MRDA corpus. However, efforts are made in order to develop a DA annotation scheme that is task-independent and can be used by automatic annotation methods [138, 139, 140]. Nevertheless, there are still limited data annotated based on the principles of these schemes, such as ISO standard 24617-2 and DIT++ [138, 139].

3.3 Proposed Model

The two parts that constitute the proposed model are depicted in Figure 3.1. The first part (sentence model) creates a vector representation of the utterance based on the LSTM structure suggested by [141] and also used by [129]. The sentence model uses word embeddings for the similarity computation between the constituent words of utterances and DA tags. This model is detailed in Section 3.3.1. The second part is a discourse model that classifies the current utterance based on its representation as well as the representations of the preceding ones as proposed by [128]. The discourse model is detailed in Section 3.3.2. To the baseline model we add the semantic representation of the DA tags.

3.3.1 Sentence Model

The proposed sentence model is an extension of the baseline sentence model with DA-specific semantic features as illustrated in Figure 3.1. The baseline sentence model and the proposed approach of semantic features extraction are described next.

Baseline Sentence Model
The baseline sentence model is depicted in Figure 3.2. Given an utterance that contains l words, the model converts it into a sequence of l d-dimensional word vectors $X_1, X_2, ..., X_l$. This sequence is given as input to the LSTM network that produces a m-dimensional vector representation s of the utterance. LSTM is a variant of RNN that has the benefit of preserving long-distance dependencies between words and distilling unimportant words from the cell gate through its forget gate layer. In particular, given a sequence $X_1, X_2, ..., X_t, ..., X_l$ of word vectors, for the t^{th} word vector X_t, with inputs h_{t-1} and c_{t-1}, h_t and c_t are computed as follows [119]:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \tag{3.1}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \tag{3.2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{3.3}$$

$$u_t = tanh(W_u x_t + U_u h_{t-1} + b_u), \tag{3.4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot u_t, \tag{3.5}$$

$$h_t = o_t \odot tanh(c_t), \tag{3.6}$$

where $W_j \in \Re^{d \times d}, U_j \in \Re^{d \times m}$ for $j \in \{i, f, o, u\}$ are weight matrices, $b_j \in \Re^d$ are bias vectors and $\sigma(\cdot)$ is the element-wise signoid function, $tanh(\cdot)$ is the hyperbolic tangent function and $\odot$ is the element-wise multiplication.

In the pooling layer, all $h_1, h_2, ..., h_t$ vectors that have been computed are combined for the generation of a single vector that represents the utterance. The combination of the h vectors can be produced by applying any of the following schemes: max-pooling, mean-pooling and last-pooling. Max-pooling keeps the element-wise maximum of the h vectors, mean-pooling averages the h vectors and last-pooling keeps the last h vector, namely the h_t vector. In order to obtain longer dependencies between the utterance words, two LSTM cells are stacked as proposed by [142] and [143]. Therefore, the sentence model has two hidden layers.

DA Representation
The typical word embeddings that constitute the input of the sentence model, does not directly model

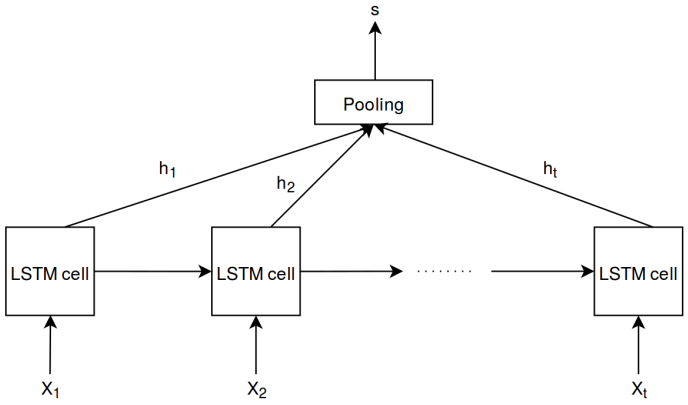


Figure 3.2: Overview of the baseline sentence model for representing utterance s.

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & d(k_1,w_1)\bar{s}(k_1,t_i) & \cdots & d(k_N,w_1)\bar{s}(k_N,t_i) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & d(k_1,w_K)\bar{s}(k_1,t_i) & \cdots & d(k_N,w_K)\bar{s}(k_N,t_i) \end{bmatrix} \cdot \begin{bmatrix} a_{i0} \\ a_{i1} \\ \vdots \\ a_{iN} \end{bmatrix} = \begin{bmatrix} 1 \\ \bar{s}(w_1,t_i) \\ \vdots \\ \bar{s}(w_K,t_i) \end{bmatrix} \tag{3.7}$$

the semantic information about the relation between each utterance word w and each DA tag. Here, we present a semantic model that automatically extracts the domain-specific semantics of w. Specifically, the semantic model computes the semantic similarity between w and each DA. The first step towards calculating semantic similarity between w and each one of the DAs, is the selection of keywords that are representative of the context of the DA tags as described in the following paragraph.

Keyword Selection. In order to automatically determine the keywords that are representative of the DAs, we use the following measurements:

1. Saliency of w, that measures the information content of w in respect to a specific task (DA in this case), as proposed by [144]:

$$L(w) = \sum_{i=1}^{T} p(t_i|w) log \frac{p(t_i|w)}{p(t_i)} \ , \tag{3.8}$$

   where $L(w)$ is the saliency of w, T is the number of DA tags, $p(t_i|w)$ is the probability of the i^{th} DA t_i given w, and $p(t_i)$ is the probability of the i^{th} DA t_i,

2. Frequency of w, denoted as $f(w)$,

3. maximum probability of a DA tag given w ($\max_{i=1}^{T} p(t_i|w)$), where t_i is the i^{th} DA.

The keyword extraction is then based on thresholds (see Section 3.5.1) applied to the product of the saliency of w and its frequency ($S(w)f(w)$) and to the maximum probability of a DA given w ($\max_{i=1}^{T} p(t_i|w)$).

Semantic Model. After determining the keywords, the semantic similarity between w and each DA is computed as follows:

$$s(w,t_i) = \sum_{j=1}^{N} a_{ij} \frac{p(t_i|k_j)p(k_j)}{p(t_i)} d(k_j,w) \ , \tag{3.9}$$

where $s(w,t_i)$ is the semantic similarity between w and the i^{th} DA t_i normalized in range 0 to 1, N is the total number of keywords and a_{ij} are the weights assigned to each keyword k_j for every

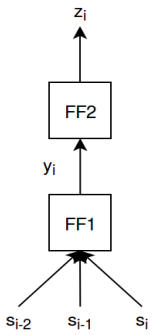


Figure 3.3: Overview of the discourse model that predicts the DA z_i of utterance s_i.

DA t_i which are computed according to (3.7) for every $i \in [1, T]$. $p(t_i|k_j)$ is the probability of the i^{th} DA t_i given the keyword k_j, $p(t_i)$ is the probability of the i^{th} DA t_i, $p(k_j)$ is the probability of the keyword k_j, $\frac{p(t_i|k_j)p(k_j)}{p(t_i)} = p(k_j|t_i)$ is the probability of being keyword k_j representative of the i^{th} DA t_i, normalized in the range 0 to 1 and $d(k_j, w)$ is the cosine similarity between the vectors of w and the keyword k_j.
In (3.7) where the a weights are calculated, K is the size of the dialogue vocabulary and $\bar{s}(w_k, t_i)$ is the estimated semantic similarity between w_k and the i^{th} DA t_i. $\bar{s}(w_k, t_i)$ is computed by applying (3.9) and setting the a weights equal to 1.

3.3.2 Discourse Model

The discourse model is depicted in Figure 3.3. Let s_i be the vector representation of the i^{th} utterance of the dialogue computed from the sentence model. The sequence s_{i-2}, s_{i-1}, s_i is used as input to a two-layer feedforward ANN. The goal of the discourse model is to predict the DA of the i^{th} utterance ($z_i \in \Re^T$). The output of the first layer of the ANN is computed as follows:

$$y_i = tanh(\sum_{d=0}^{2} W_{-d} s_{i-d} + b_1), \tag{3.10}$$

where $W_0, W_{-1}, W_{-2} \in \Re^{T \times m}$ are the weight matrices, $b_1 \in \Re^T$ is the bias vector, $y_i \in \Re^T$ is the DA representation of the s_i utterance, and T is the number of DAs.

Next, the input of the second layer of the ANN is the vector representation y_i provided by the first layer. The final output of the network is the prediction of the DA for the utterance s_i computed as follows:

$$z_i = softmax(U_0 y_i + b_2), \tag{3.11}$$

where $U_0 \in \Re^{T \times T}$ and $b_2 \in \Re^T$ are the weight matrices and bias vector, respectively. For the discourse model, history size of two previous utterances is used for the first layer and no history is taken into account for the second layer as recommended by [128].

3.4 Experimental Dataset

The dataset used is the Switchboard-DAMSL dataset [42], which is annotated with the 42 DAMSL tags. The Switchboard corpus was originally used for training and testing various speech processing algorithms. Also, it has been used for other tasks such as Automatic Speech Recognition (ASR) [145] and acoustic model adaptation [146], including the modeling of DAs [131]. This dataset is split into training and test subsets as proposed by [124]. The training set comprises of 1,155 dialogues (199,050 utterances) and the test set of 19 dialogues (3,927 utterances) collected over the phone from 500 different speakers. The word-by-word transcriptions are also provided. The topic of discussion

between two speakers is introduced by a computer-driven robot agent and the conversation that follows is recorded. About 70 casual topics were introduced. In Table 3.1, the length of the dialogues (in terms of number of utterances) included in the dataset is presented. A development set was created

# of Utterances per dialogue	Train set	Test set
min value	92	187
max value	954	679
mean value	334.6	410.0

Table 3.1: Switchboard-DAMSL corpus.

by randomly selecting 115 dialogues (13,192 utterances) from the training set.

In Table 3.2, representative examples of the eight most frequent DAs are presented. Furthermore, the distribution of the DAs over the dataset is reported in Table 3.3. As shown in this table, the most frequent DA is the "Statement-non-opinion".

No preprocessing, including tools for stripping the punctuation and changing the capitalization, is applied to the dataset. For the experiments that follow classification accuracy is used as evaluation measurement.

DA tag	Example
Statement-non-opinion	There's no one else that works there.
Acknowledge (Backchannel)	Sure.
Statement-opinion	but I think its relevance is pretty limited.
Agree/Accept	That's right.
Abandoned or Turn-Exit	Do you,-
Appreciation	Well good.
Yes-No-Question	So do you have a family too?
Non-verbal	<Laughter>.

Table 3.2: Examples of the most frequent DAs.

3.5 Parameter Tuning

In this point, we describe the process for selecting the keywords of the semantic model (see Section 3.5.1) and the tuning of the hyperparameters of the LSTM baseline model (see Section 3.5.2). For tuning we used the development set mentioned in Section 4.5.

3.5.1 Keyword Selection

For the selection of the keywords, classification accuracy is calculated when different thresholds to the metrics described in Section 3.3.1 are applied. The best performance is achieved when 323 keywords are selected (for $S(w)f(w) = 200$ and $\max_{i=1}^{T} p(t_i|w) = 0.5$). Indicative examples of the selected keywords for the most frequent DAs are presented in Table 3.4.

DA tag	Train set (%)	Test set (%)
Statement-non-opinion	36.9	31.5
Acknowledge (Backchannel)	18.8	18.2
Statement-opinion	12.7	17.1
Agree/Accept	7.6	8.6
Abandoned or Turn-Exit	5.5	5.0
Appreciation	2.3	2.2
Yes-No-Question	2.3	2.0
Non-verbal	1.7	1.9
Remaining DAs	*12.2*	*13.5*

Table 3.3: Relative frequency (%) of the DAs.

DA tag	Selected keywords
Statement-non-opinion	want, can't, work, mine, decided, always, remember
Acknowledge (Backchannel)	huh-uh, huh, yeah, yep, what?, huh?
Statement-opinion	seem, think, scary, ought, worse, difficult
Agree/Accept	true, agree, yes
Abandoned or Turn-Exit	–, -/, -
Appreciation	gosh, dear, wow, kidding
Yes-No-Question	mean?, there?, then?, all?
Non-verbal	<Laughter>, <Noise>, <Clicking>., <sniffing>

Table 3.4: Examples of automatically selected keywords (shown for most frequent DAs).

3.5.2 LSTM Parameters

For the implementation of the baseline sentence model (see Section 3.3.1) the NN packages provided by [141] and [147] were used. One hyperparameter at a time is tuned while keeping the remaining ones fixed in order to determine the best configuration. Based on findings taken from literature [129], we initialize the parameters with the following values: word embeddings=200-dimensional vectors with GloVe [110], decay rate=0.7, dropout=0.3, pooling-mechanism=mean-pooling.

Word Embeddings. Keeping the hyperparameters of the LSTM network fixed, different word-to-vector techniques and the dimensionality of the word vectors, that constitute part of the input to the network, are tested. The word vectors are trained either with word2vec [120, 109] method on the GoogleNews corpus or with the GloVe [110] method on the CommonCrawl corpus. Regarding the dimensions of the word embeddings, we use those referred in [128] [1]. The word embeddings

[1] The word2vec method yields lower classification accuracy (by 0.2%) compared to GloVe and is not reported in Table 3.5.

Word embeddings	Decay rate	Dropout	Pooling mechanism	Classification Accuracy(%)
50	0.7	0.3	mean	74.7
150				75.4
200				**75.6**
300				75.2
200	0.3	0.3	mean	74.3
	0.5			75.1
	0.9			74.1
200	0.7	0.0	mean	75.4
		0.1		75.4
		0.2		75.5
		0.4		75.4
		0.5		75.2
200	0.7	0.3	max	75.3
			last	75.2

Table 3.5: Performance of LSTM hyperparameters w.r.t. test set.

are then concatenated with the features extracted by the semantic model. The performance for various dimensions is presented in Table 3.5. As shown in this table, the best performance (75.6%) is achieved when 200-dimensional word embeddings are used. Therefore, for the experiments that follow this setting is used.

Decay Rate. The decay rate is a regularization factor of the update of the network connection weights in order to avoid overfitting of the network. Typically, the decay rate value lies between 0 and 1. In this work, the decay rates that are recommended in the literature [128, 129] are examined, as shown in Table 3.5. The best performance (75.6% accuracy) is achieved with decay rate equal to 0.7 and this setting is used for the rest experiments.

Dropout. For most DNNs, dropout [148] is used as a regularization technique against overfitting. In Table 3.5, the impact of dropout rate on the classification accuracy is presented for values in the range between 0.0 and 0.5 as proposed in the literature [128, 129]. The best performance (75.6% accuracy) is achieved when the dropout rate equals to 0.3 and this setting is used for the experiments that follow.

Pooling mechanism. The various mechanisms that can be used in the pooling layer (max-, mean-, and last-pooling) as described in Section 3.3.1, are tested. The performance (classification accuracy) for various pooling schemes (max, mean, last) is reported in Table 3.5. The highest classification accuracy (75.6%) is yielded by the mean-based scheme, which is adopted.

Other Hyperparameters. Here, we briefly mention the settings for a number of other parameters following literature findings [129]. The value of $l2$-regularization is set at $1e-5$ and the $tanh$ function is used for activation in the LSTM cell. Moreover, as reported by [129] changes on the learning rate do not have an impact on the performance of the model. Hence, the learning rate is set at $1e-3$.

3.5.3 Evaluation Results

In Table 3.6, the classification accuracy for both the baseline and proposed model is reported. The highest accuracy (75.6%) is achieved by the proposed model outperforming the baseline by 3.8% when both sentence and discourse information is used. Regarding the sentence-level analysis, the difference between the proposed model and the baseline is even bigger (4.3%). In Table 3.6 the performance of the baseline model, when applying preprocessing of the dataset, is also presented. In this case, the proposed model still outperforms the baseline by 1.7% accuracy.

Model	Analysis Level	Preprocessing	Classification Accuracy(%)
Baseline	sentence	✗	69.5
		✓	72.8
Proposed		✗	**73.8**
Baseline	Sentence & discourse	✗	71.8
		✓	73.9
Proposed		✗	**75.6**

Table 3.6: Performance of the baseline and the proposed model.

Model	Classification Accuracy(%)
Majority classification baseline	*31.6*
Proposed	**75.6**
HMM [124]	71.0
LSTM [128]	69.6
CNN [128]	73.1
RCNN [127]	73.9
DRLM-joint training [125]	74.0
DRLM-conditional training [125]	**77.0**
Tf-idf (baseline)	47.3
Inter-annotator agreement	*84.0*

Table 3.7: Performance of the proposed model and other methods from the literature.

Based on the results of Table 3.6, the proposed model benefits from the additional semantic information. Moreover, it is demonstrated that the proposed model avoids the need for preprocessing of the dataset[2].

The performance of the proposed model is comparable with the state-of-the-art[3] classification accuracy (see Table 7 for an overview) which equals to 77.0% [125]. An advantage of the present work is the utilization of straightforward feature extraction compared to [125] that requires the identification of latent discourse-level features.

3.5.4 Conclusions

In this work, we demonstrated the effectiveness of the incorporation of DA-specific semantic features in RNN-based DA classification. Those features were computed with respect to a set of salient keywords meant to semantically represent the DA of interest. The proposed features were found to yield 1.7% (absolute) improvement in classification accuracy with respect to the baseline approach that relies solely on word-level embeddings. Also, we experimentally showed that the discourse-level (specifically, the consideration of current and the previous two utterances) further improves on the baseline performance. Unlike similar approaches presented in the literature, the proposed model does not require any additional tools meant for the preprocessing of dialogues transcriptions.

[2] This was experimentally justified, so, the performance of the proposed model when applying data preprocessing is not reported.

[3] Also, we replicated (use of same model implementation and data) the experiments proposed in [129] without achieving the same results.

Regarding future work, we plan to investigate the incorporation of more features derived from deeper discourse analysis. In addition, we aim to further validate the experimental findings of this work by using datasets in languages other than English.

Chapter 4

Transfer Learning from Intent and Emotion to Document-level Personality Recognition

4.1 Introduction

According to trait theory, individuals can be characterized in terms of relatively enduring patterns of thoughts, feelings and actions. The task of personality recognition aims to categorize individuals among a set of quantitatively assessed traits. Automatic personality recognition has many real life applications. In recommendation systems, suggestions about an individual can derive from the preferences of others with similar personality [149, 150]. In human-machine interaction, the performance of dialogue systems can be improved by enhancing them with personality reading capabilities [151, 152].

Previous successful approaches on personality recognition from documents leverage lexical information using both traditional and deep learning architectures [4, 5, 6]. State-of-the-art models also incorporate psycholinguistic and affective features, typically extracted from knowledge bases or lexicons, at the document level. However, such late-fusion schemes only provide coarse-grained information to the model. Moreover, there are other information sources that although are considered indicative of an individual's personality have yet to be exploited. In particular, previous studies suggest the association of individuals' personality with the context of a given situation [3, 2, 153, 154] including their intentions.

Transfer Learning (TL) aims at using knowledge from related source domains in order to improve the performance on a target task. Neural TL, which is a common practice in Computer Vision (CV) [155, 156], has recently shown promising results in NLP [157, 158]. TL offers an alternative way for incorporating affective and intent information to automatic personality recognition. Specifically, low-level and fine-grained information learnt by deep neural networks can be transferred to personality recognition models in an automatic and scalable way.

In this work, we introduce two TL methods for exploiting information from pretrained sentence-level models for document-level personality recognition. Our models are based on Hierarchical Attention Networks (HAN) [159]. First, we train a model on tasks related to the target domain, namely emotion and intent recognition. Next, we utilize the encoder of the pretrained network for initializing the sentence encoder of the HAN and fine-tune the model on the target task. We also use the same pretrained encoder as a feature extractor in order to enrich the sentence-level representations of the network. Further, as proposed in the literature [5, 160], we investigate the incorporation of lexicon-based psycholinguistic features to the model. The proposed TL approaches are evaluated on the YouTube [161] and the stream-of-consciousness essays [162] datasets and achieve state-of-the-art results.

The main contributions of the paper are:

- We propose novel adaptations of two TL methods to transferring knowledge from sentence-level source tasks to a document-level problem, achieving state-of-the-art results.
- We conduct analysis of the contribution of different types of information, namely intent, emotion, psycholinguistic) to the personality recognition task. This is the first time to our knowledge that intent was succesfully applied to the problem of personality recognition.

Big Five Trait	Description
Extraversion (Extr)	outgoing/energetic vs. solitary/reserved
Agreeableness (Agr)	friendly/compassionate vs. challenging/detached
Conscientiousness (Cons)	efficient/organized vs. easy-going/careless
Neuroticism (Neu)	sensitive/nervous vs. secure/confident
Openness (Open)	inventive/curious vs. consistent/cautious

Table 4.1: Big Five model description

The paper is structured as follows. In Section 4.2, prior work is presented. In Section 4.3, the baseline model based on a hierarchical attention RNN is outlined and in Section 4.4 the proposed approach is described in detail. In Section 4.5, the experimental datasets, setup and results are provided and Section 4.6 concludes this work.

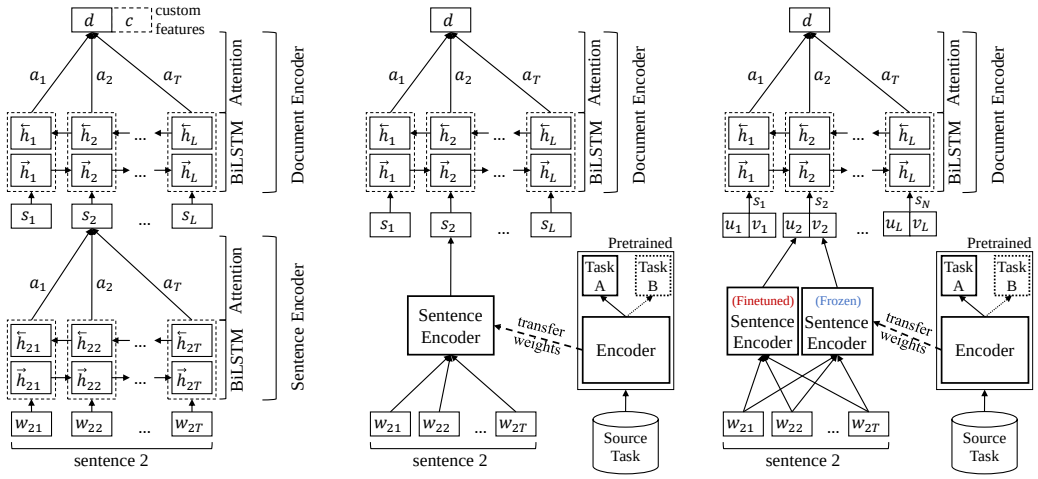
4.2 Related Work

Personality Recognition. The most popular psychological model for describing an individual's personality is the Big Five model (or Five Factor Model (FFM)), as described in Table 4.1. Previous approaches in personality recognition focus on the utilization of psycholinguistic and emotion lexicons for predicting personality traits of users in social media [163, 164, 160]. Specifically, [67] utilized the Linguistic Inquiry Word Count (LIWC) lexicon [165], that contains psycholinguistic information. [166] tested different subsets of these features in order to further boost the performance and [167] added to the feature space other relevant features, such as imageability, in order to predict the personality traits of authors. [168, 4] exploited additional lexical resources such as the NRC [169], MRC [170], SentiStrength [1] and SPLICE lexica [2]. [5] tested the performance of such features using different algorithms in social media datasets (twitter, facebook, youtube). Also, they investigated and took advantage of the correlation between the personality dimensions, reporting state-of-the-art results in social media datasets.

Recently, deep learning approaches have been successfully applied to predicting personality in large datasets of short texts (e.g. tweets, facebook statuses) [167, 171]. [6] introduced a deep learning approach in document-level personality recognition of authors based on hierarchical Convolution Neural Networks (CNNs). However, due to limited annotated data the performance of the hierarchical CNN did not exceed those of previous approaches. For this reason, [6] filtered the dataset based on the NRC emotion lexicon and utilized the LIWC lexicon in order to construct custom features (Mairesse features [67]). Moreover, in similar problems related to document classification tasks, such as sentiment estimation and topic classification, Hierarchical Attention Networks (HAN) [159] have been successfully used.

Transfer Learning (TL). TL aims at making use of the knowledge from a source domain, to improve the performance of a model in a different, but related, target domain. Neural TL has been applied with great success in Computer Vision (CV) [155, 156]. Deep neural networks in CV are rarely trained from scratch and instead are initialized with pretrained models. Notable examples include face recognition [172] and visual Question Answering (QA) [173], where image features trained on ImageNet [174] and word embeddings learnt on large corpora via unsupervised training are combined. Although model transfer has seen widespread success in CV, neural TL beyond pretrained word vectors

[1] http://sentistrength.wlv.ac.uk.
[2] http://splice.cmi.arizona.edu.



(a) Baseline HAN **(b)** INIT-TL model **(c)** HTL model

Figure 4.1: Overview of the proposed models

is less pervasive in Natural Language Processing (NLP). Some recent attempts have demonstrated the successful utilization of neural Language Models (LM) as generic feature extractors [158]. TL has also been adopted to sentence-level classification tasks, such as emotion recognition [175], sentiment analysis and sarcasm detection [157], by pretraining the respective models on semantically similar tasks. However, such methods have not yet been tested in the personality recognition domain.

4.3 Baseline

The baseline model is a Hierarchical Attention Network (HAN), as depicted in Figure 4.1a. We train a separate model for each personality dimension. The input to the network is a document consisting of L sentences, where each sentence contains T words.

Sentence Encoder (SE). We use a word embedding layer to project the words $w_1, w_2, ..., w_T$ to a continuous vector space R^E, where E the size of the embedding layer. Afterwards, we encode the information in each sentence using Long Short-Term Memory (LSTM) [176] networks. An LSTM takes as input the word embeddings in a sentence and produces word annotations $h_1, h_2, ..., h_T$, where h_i is the hidden state of the LSTM at time-step i, summarizing all the information of the sentence up to the i-th word. We use Bidirectional LSTM (BiLSTM) in order to get word annotations that summarize the information from both directions. A BiLSTM consists of a forward LSTM $\overrightarrow{f}$ that reads the sentence from w_1 to w_T and a backward LSTM $\overleftarrow{f}$ that reads the sentence from w_T to w_1. We obtain the final annotation for a given word w_i, by concatenating the annotations from both directions, $h_i = \overrightarrow{h_i} \parallel \overleftarrow{h_i}, \quad h_i \in R^{2S}$ where $\parallel$ denotes the concatenation operation and S the size of each sentence-level LSTM.

On top of the LSTM network, an attention mechanism is added in order to identify the most informative words in each sentence. The attention mechanism assigns a weight a_i to each word annotation h_i. We compute the fixed representation s of each sentence as the weighted sum of all the word annotations.

$$e_i = tanh(W_h h_i + b_h), \quad e_i \in [-1, 1] \tag{4.1}$$

$$a_i = \frac{exp(e_i)}{\sum_{t=1}^{T} exp(e_t)}, \quad \sum_{i=1}^{T} a_i = 1 \tag{4.2}$$

$$s = \sum_{i=1}^{T} a_i h_i, \quad s \in R^{2S} \tag{4.3}$$

where W_h and b_h are the attention layer's weights.
Document encoder. The document encoder uses the same architecture as the sentence encoder, excluding the embedding layer. It reads the produced sequence of sentence representations $s_1, s_2, ..., s_L$ and generates a final fixed feature representation d for each document.
Output Layer. We feed d to a final task-specific layer with one neuron for performing classification or regression, per personality dimension. When adding document-level custom features, as depicted in Figure 4.1a the final feature vector is computed as $d \parallel c$, where $\parallel$ denotes the concatenation operation and c is the custom feature vector.

4.4 Transfer Learning

We make use of knowledge from source tasks related to personality recognition, in order to improve the performance on the target task. It is more likely that the source and the target data originate from the same distribution. Presumably, the features learned from the source task will be useful for modeling the samples in the target task.

We adapt two well-known TL methods to transferring knowledge from sentence-level tasks to our document-level problem, which is modeled with a hierarchical attention RNN. Specifically, we train a SE separately on intent and emotion recognition as well as on both tasks via multi-task learning. We propose two different TL methods for utilizing the knowledge contained in the pretrained SE.

4.4.1 Pretraining the SE

In order to pretrain the SE, we utilize the intent and emotion recognition tasks. As suggested in literature, action and emotion concepts are highly correlated with personality and in particular with specific traits [153, 154]. We also experiment with combining both types of information via multi-task learning.

In the case of multi-task learning, the loss is computed as the weighted sum of the two individual losses, as follows:

$$L = L_a + \alpha L_b \tag{4.4}$$

where L_a is the intent loss, L_b is the emotion loss and α is the multiplier used for mitigating the loss imbalances.

4.4.2 Initialization (INIT-TL) of SE

The most common way for transferring knowledge is to fine-tune a pretrained model on the target task. Since we aim at transferring knowledge from a sentence-level to a document-level task, we initialize the weights of the SE with our pretrained model, whereas the document encoder is randomly initialized. Next, we fine-tune both encoders on the target task [3]. The proposed architecture is presented in Figure 4.1b.

4.4.3 Hypercolumns-TL (HTL) of SE

An alternative way to transfer knowledge from a source domain to a target task is by extracting features from a pretrained neural network, known as hypercolumns in CV [177]. Concretely, we use two separate sentence encoders. One of them is randomly initialized and fine-tuned on the target task. The weights of the other SE are initialized from the encoder of the pretrained model and remain frozen during training, in order to preserve the knowledge from the source task. This also acts as an extra

[3] We experimentally found that the proposed model achieves better results when both encoders are fine-tuned instead of keeping the SE frozen.

safety measure against catastrophic forgetting. We obtain the augmented sentence representations s_i, which is fed to the document encoder, as follows:

$$s_i = u_i \parallel v_i \tag{4.5}$$

where u_i and v_i are the sentence representations of task-specific and pretrained SEs, respectively. The above approach is presented in Figure 4.1c.

4.5 Experiments & Results

In this section we present the dataset used for the pretraining of the SE as well as the two datasets, on which we evaluate our proposed TL approaches for personality recognition. We also describe our experimental setup and demonstrate our results, alongside an analysis of the contribution of each information source, i.e. emotion, intent, psycholinguistics.

4.5.1 Experimental Datasets

In order to evaluate our proposed models on personality recognition, we use two datasets of different nature. One derives from social media and is annotated on a continuous scale and the other consists of written essays and is annotated on a binary scale. Finally, for pretraining (see Section 4.4.1) we use a much larger dataset annotated with intent and emotion labels.

Pretraining dataset: DailyDialog

The DailyDialog dataset [178] consists of **13,118** dialogues across different topics, such as everyday life, school life, culture and education and politics. Each dialogue of the dataset consists of approximately 8 speaker turns with 14.6 tokens per speaker turn on average. The dataset is annotated in terms of both intent tags (inform, question, commissive, directive) and discrete emotions (no emotion, anger, disgust, fear, happiness, sadness, surprise) for each speaker turn of each dialogue. In Tables 4.2 and 4.3 we present the distribution of the labels for intent and emotion in the dataset, respectively.

Intent dimension	**Absolute Value**	**Percentage (%)**
Inform	46,532	45.2
Question	29,428	28.6
Directives	17,295	16.8
Commissive	9,724	9.4

Table 4.2: Intent statistics in DailyDialog

Emotion dimension	**Absolute Value**	**Percentage (%)**
Anger	1,022	0.99
Disgust	353	0.34
Fear	74	0.17
Happiness	12,885	12.51
Sadness	1,150	1.12
Surprise	1,823	1.77
No emotion	85572	83.10

Table 4.3: Emotion statistics in DailyDialog

Model	Extr (%)	Agr (%)	Cons (%)	Neu (%)	Open (%)
Majority	51.72	50.02	53.10	50.79	51.52
State-of-the-art (CNN, NRC & Mairesse features) [6]	58.09	59.38	56.71	57.30	62.68
HAN baseline	55.73	56.50	55.21	55.33	61.68
HAN baseline & LIWC	58.71	59.76	58.47	**58.71**	**64.14**
INIT-TL (intent) (FT)	57.21	57.54	57.94	58.19	62.84
INIT-TL (emotion) (FT)	57.70	60.05	58.47	56.36	61.91
INIT-TL (intent & emotion) (FT)	58.31	60.00	**59.56**	58.18	63.37
HTL (intent)	**58.80**	58.87	58.21	**58.70**	63.49
HTL (emotion)	57.54	**60.58**	58.54	58.14	**64.14**
HTL (intent & emotion)	57.90	60.13	58.75	58.51	62.15

Table 4.6: Results on the stream-of-consciousness essay dataset. The scores are computed using the *accuracy* metric using 10-fold cross-validation.

YouTube dataset

The YouTube dataset [161] consists of **404** vlogs, each one by a different vlogger. The scores assigned to each vlogger for each one of the five personality traits are impression scores set by viewers. After averaging the scores assigned for each vlog, the annotations for each personality trait lie in the interval [1, 7] with real values. Moreover, the YouTube dataset offers a set of extracted audiovisual features. In Tables 4.4, the distributions of the number of words per sentence and the number of sentences per vlog are presented, respectively.

	min	max	avg
words per sentence	1.00	137.00	14.72
sentences per vlog	2.00	141.00	37.65

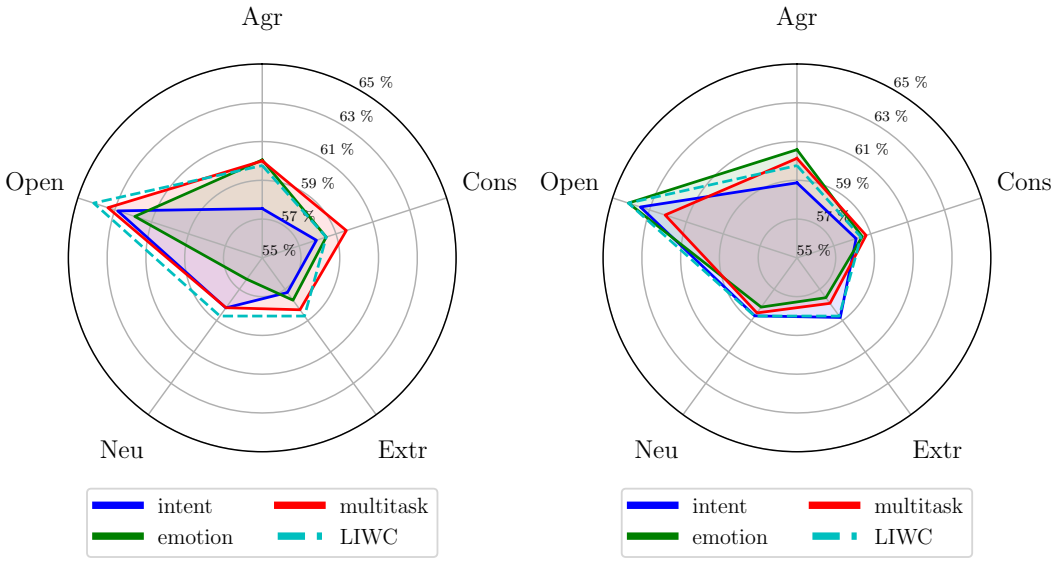
Table 4.4: Statistics for the YouTube dataset

Stream-of-consciousness essay dataset

The stream-of-consciousness essay dataset provided by [162] contains **2,468** anonymous essays tagged with the authors' personality traits. The stream-of-consciousness essays were written by volunteers in a controlled environment and their authors were asked to define their own Big Five personality traits. The personality traits for each author have been annotated with a "yes" or "no" tag. In Table 4.5, the distributions of the number of words per sentence and the number of sentences per essays are presented, respectively.

	min	max	avg
words per sentence	1.00	892.00	13.35
sentences per essay	1.00	336.00	28.60

Table 4.5: Statistics for the essays-of-consciousness dataset



(a) INIT-TL **(b)** HTL

Figure 4.2: Contribution of information sources for each TL approach in stream-of-consciousness essays dataset. The evaluation metric is *accuracy*, which means that larger surfaces denote better performance. The proposed approaches are competitive with the HAN model with the LIWC features.

Model	**Extr (RMSE)**	**Agr (RMSE)**	**Cons (RMSE)**	**Neu (RMSE)**	**Open (RMSE)**
State-of-the-art (MTSC) [5]	0.85	**0.72**	0.69	**0.70**	0.69
HAN baseline	0.80	0.87	0.74	0.78	0.74
HAN baseline & LIWC	**0.77**	0.76	0.71	0.73	0.68
INIT-TL (intent) (FT)	0.78	0.77	0.72	0.75	0.68
INIT-TL (emotion) (FT)	0.79	**0.72**	**0.68**	0.71	0.68
INIT-TL (intent & emotion) (FT)	0.78	0.73	0.70	0.72	0.67
HTL (intent)	**0.77**	0.76	0.71	0.74	0.69
HTL (emotion)	**0.77**	0.74	0.72	0.71	**0.65**
HTL (intent & emotion)	0.78	0.73	0.69	**0.70**	0.67

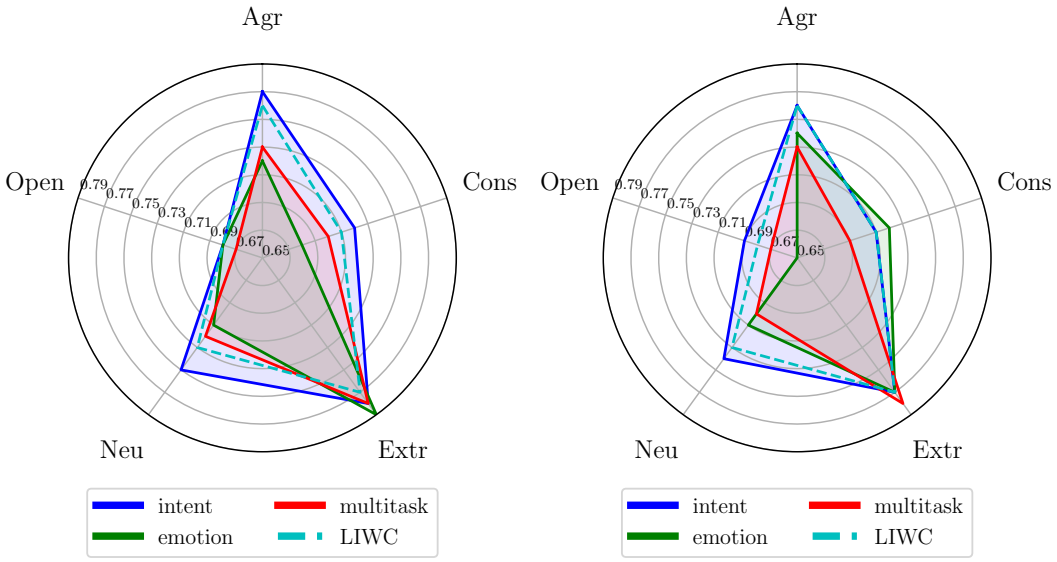
Table 4.7: Results for the YouTube dataset. The scores are computed using the *RMSE* metric, after a 10-fold cross-validation. Custom audiovisual features are included in all models (see Figure 4.1a).

4.5.2 Experimental Setup

We initialize the embedding layer of our models with pretrained 300-dimensional GloVe word embeddings [110] [4]. We also add Gaussian noise with $\sigma = 0.1$ to the embedding layer, which can be interpreted as a random data augmentation technique, that makes our models more robust to overfitting. Moreover, we apply dropout [179] of 0.2 to the embedding layer and we use early-stopping. Furthermore, we fine-tune the weights of the embedding layer during training in all tasks and we utilized the Adam optimizer [115]. Finally, we use one BiLSTM layer with 150 neurons (per direction), for both the sentence and the document encoder.

We use a different loss function for training each of the previously described models. For the pretrained models on intent and emotion recognition, we utilize the cross entropy loss function. In the

[4] We used the 6B GloVe embeddings to reduce memory footprint.



(a) INIT-TL **(b)** HTL

Figure 4.3: Contribution of information sources for each TL approach in YouTube dataset. The evaluation metric is *RMSE*, which means that smaller surfaces denote better performance. Both approaches outperform the HAN model with the LIWC features for most settings.

case of multi-task learning, we empirically set the value of the a multiplier in Equation 4.4 to 1.5. For training the HAN on personality recognition in the stream-of-consciousness essay dataset we use the binary cross entropy loss and in the YouTube dataset the *Mean Square Error* (MSE) loss function.

In order to pretrain the SE on the DailyDialog dataset, we adopt the standard train-validation-test split, as provided by [178]. The evaluation of our models on personality recognition is based on 10-fold cross-validation as already proposed in the literature [5, 6] for these datasets. Specifically, for the YouTube dataset, we use the *Root Mean Squared Error* (RMSE) evaluation metric, since we address a multi-label regression problem. For the stream-of-consciousness essays dataset, which is a multi-label binary classification problem, we use the *Accuracy* metric in accordance with the literature.

Finally, in a subset of the conducted experiments, we also utilize lexicon-based LIWC features, which are commonly used in the literature [5, 166, 6]. Specifically, we extract for each document (vlog or essay depending on the used dataset) 81 psycholinguistic features at document level. These features are related to dimensions such as social processes (e.g. "talk", "mate", "child"), affective processes (e.g. "happy", "cried", "abandon"), anxiety (e.g. "worried", "nervous") and cognitive processes (e.g. "cause", "know"). A complete overview of all psycholinguistic dimensions included in the LIWC lexicon is provided by [180]. In the case of the YouTube dataset, we also utilize in all experiments 26 custom audiovisual features provided by [161]. These custom features are added to the models as depicted in Figure 4.1a (subset of custom feature vector c).

4.5.3 Results

TL methods. In Tables 4.6 and 4.7, we present the performance of the two proposed TL methods, namely INIT-TL and HTL, on stream-of-consciousness essays and YouTube datasets respectively. We also provide the state-of-the-art results as well as the performance of our baseline model when adding the lexicon-based LIWC features. For each TL method we report the experimental results when pretraining the SE on intent, emotion and both tasks via multi-task learning.

As observed, both TL approaches improve the performance over the baseline model. Specifically, for the stream-of-consciousness essays dataset the improvement in the results can reach up to 4% in (absolute) accuracy (*Agr* trait), depending on the utilized pretrained model. In the YouTube dataset,

the best results are also achieved in the *Agr* trait, where the HTL method with a SE pretrained on emotion can provide an improvement of 0.15 (RMSE metric). Overall, both INIT-TL and HTL present comparable performance across all traits and datasets. However, in most experimental setups the HTL method achieves better results in comparison with the INIT-TL. In fact, for specific personality traits and pretrained SEs, the HTL method outperforms the INIT-TL by up to 1% in (absolute) accuracy for the essays dataset (Table 4.6, INIT-TL vs. HTL with intent for *Agr* trait). For the YouTube dataset (Table 4.7), the performance difference between the two TL methods is smaller.

Both TL approaches provide results comparable with those of the baseline HAN when adding custom LIWC features (state-of-the-art approach). In particular, for the stream-of-consciousness essays dataset (Table 4.6), in the personality dimensions of *Extr*, *Agr* and *Cons* the HTL method performs better than the baseline HAN enhanced with the LIWC features, achieving accuracy of **58.80%**, **60.58%** and **59.56%** respectively. In the remaining dimensions, namely *Neu* and *Open*, the performance of the two approaches is comparable (**58.70%** and **64.14%**, respectively when utilizing different pretrained SE in the case of the HTL method). For the YouTube dataset (Table 4.7), our approach outperforms the baseline HAN with LIWC features for all dimensions, apart from *Extr* where the performance of the two approaches is comparable (**0.77** RMSE).

Information sources. As a further investigation step, we visualize the performance of the proposed methods and analyze the contribution of each information source across all TL methods and datasets. We also compare the proposed models against the HAN baseline with the addition of the LIWC features (HAN-LIWC). In Figure 4.2, we present the results of the stream-of-consciousness essays dataset. The evaluation metric is accuracy, which means that larger surfaces are better. We observe that both TL methods are comparable with HAN-LIWC. Figure 4.3 depicts the results in the YouTube dataset. Since the evaluation metric is RMSE, a smaller surface denotes better performance. In this case, we observe that both approaches improve on the state-of-the-art and outperform HAN-LIWC.

We can visually verify that HTL method outperforms the INIT-TL method in most settings, also providing more robust results. In addition to validating that psycholinguistic features (LIWC) improve the prediction power of our model, we substantiate the assumption that both intent and emotion information contribute to the prediction of an author/vlogger's personality traits. When comparing the performance of our models on the two datasets, we observe that intent information contributes significantly to predicting the authors' personality (stream-of-consciousness essays dataset), while for the YouTube dataset emotion is the dominant information source.

Moreover, we observe that the incorporation of both emotion and intent information via the multi-task pretraining, achieves the most consistent results. For the stream-of-consciousness essays dataset the two types of information seem to act complementary to each other; this is more evident for the INIT-TL method (Figure 4.2a). However, for the HTL method the performance gains are reduced and in some cases the addition of individual information sources outperforms the combination of both. This can be attributed to the fact that by keeping the pretrained encoder frozen, we overcome the effects of catastrophic forgetting, by preserving the knowledge from the source task. This effect is more evident in cases where there is strong correlation between a personality trait and a specific information source. Whereas the relevant information is diluted in the multi-task pretraining process.

4.6 Conclusions

In this work we explore document-level personality recognition in a Transfer Learning (TL) framework. We propose two TL methods to make use of sentence-level affective and intent information, improving model performance. Models are evaluated in YouTube and stream-of-consciousness essays datasets, yielding state-of-the-art results. Analysis of our results shows that affective, intent and psycholinguistic information contribute to personality recognition. Different types of information are more dominant for specific traits and datasets. Combination of affective and intent information in a multi-task setting provides further increase in performance, especially in the case of essays dataset.

In future work, we plan to investigating more sophisticated TL approaches for transferring knowl-

edge from sentence to document-level tasks. We will also further investigate the combination of the proposed TL methods with lexicon-based features for personality recognition and incorporate additional sources of information.

Chapter 5

Conclusions

In this work, we investigate deep learning methods for predicting intent and personality recognition of speakers. Both models can be incorporated in a Dialogue System for the improvement of human-machine interaction. Specifically, recognizing the Dialogue Acts (DAs) of each utterance of the speaker contributes to the generation of an appropriate response by the DS. By understanding the current intentions of the speaker (e.g. inform, question, feedback, backchannel etc.), DSs can restrict the number of options produced as possible responses. While, intent recognition can improve the human-machine interaction, recognizing an individual's personality can contribute to more personalized DSs. If a DS is able to successfully identify the personality traits of each user, it will consequently adapt its behavior. Moreover, specific personality traits can also be adopted by the DS in order to attribute a specific "personality" for the system. In the context of this thesis, we proposed two different models for predicting the intentions and personality traits of speakers, respectively.

First, we implemented a deep learning model for classifying the Dialogue Acts (DAs) of each utterance in the context of dialogues. Our model consist of a LSTM network for producing the representation of each utterance and, next, a simple Feed-Forward Neural Network (FFNN) for classifying the current utterance based on its representation as well as those of the two previous utterances of the dialogue. The FFNN aims at exploiting the history of the dialogue before predicting the DA of the utterance. Our contribution to this model is the expansion of the generic word embeddings that are fed to the network. Specifically, we incorporate DA-specific information in the generic word embeddings. We test the proposed method using the Switchboard-DAMSL corpus, which is annotated with 42 DA tags and achieve results comparable with the state-of-the-art.

Second, we propose two Transfer Learning (TL) methods for document-level personality recognition. Our motivation for using TL is that the personality traits are correlated with other information, such as emotion, intent and psycholinguistic. In fact, to the best of our knowledge, we first introduce intent as a related information source to the problem of personality recognition. Previous approaches in personality recognition have utilized such related information to improve the performance of their models in the form of handcrafted features and lexicons. We introduce a novel adaptation of two well-known TL methods for transferring knowledge from sentence-level source tasks to the document-level target task. Our model is based on hierarchical attention networks. We train a model on intent, emotion or both via multi-task learning. Next, we utilize the encoder of the pretrained model for initialization of the sentence-encoder of our model as well as a sentence-level feature extractor. We evaluate our approach in two personality datasets, namely the YouTube and stream-of-consciousness essay datasets and achieve state-of-the-art results. Furthermore, we conduct an analysis on the contribution of different information sources (i.e. emotion, intent, psycholinguistic) and prove that all information can improve the prediction of personality. A different type of information is more dominant for each personality trait and dataset.

In the future, we can improve the DA classification model, by replacing the FFNN, utilized for exploiting information related to dialogue history, with a LSTM model for preserving long short-term information of sequential utterances. We also evaluate the proposed approach in other datasets commonly used for DA classification. Furthermore, for the personality recognition problem, we can experiment with more information sources, except for emotion, intent and psycholinguistics, for pretraining the sentence encoder of our network. Moreover, we can test the incorporation of the lexicon-based psycholinguistic features to our TL approach.

Finally, the individual suggested models can be combined in terms of a DS alongside other possible features (e.g. empathy or engagement recognition). We can also implement a generative language model conditioned on these features for producing appropriate responses. This will help the personalization and adaptation of the DSs on the individual users. The contribution of such features to the performance of a DS can be evaluated by comparing it with the respective baseline system.

Bibliography

[1] J. M. Digman and J. Inouye, "Further specification of the five robust factors of personality." *Journal of personality and social psychology*, vol. 50, no. 1, p. 116, 1986.

[2] R. R. McCrae and P. T. Costa Jr, "A five-factor theory of personality," *Handbook of personality: Theory and research*, vol. 2, pp. 139–153, 1999.

[3] W. Mischel, "Toward a cognitive social learning reconceptualization of personality." *Psychological review*, vol. 80, no. 4, p. 252, 1973.

[4] C. Sarkar, S. Bhatia, A. Agarwal, and J. Li, "Feature analysis for computational personality recognition using youtube personality data set," in *Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition*. ACM, 2014, pp. 11–14.

[5] G. Farnadi, G. Sitaraman, S. Sushmita, F. Celli, M. Kosinski, D. Stillwell, S. Davalos, M.-F. Moens, and M. De Cock, "Computational personality recognition in social media," *User modeling and user-adapted interaction*, vol. 26, no. 2-3, pp. 109–142, 2016.

[6] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 74–79, 2017.

[7] J. R. Searle, *The construction of social reality*. Simon and Schuster, 1995.

[8] M. Tomasello and M. Carpenter, "Shared intentionality," *Developmental science*, vol. 10, no. 1, pp. 121–125, 2007.

[9] D. Jacquette, "Brentano's concept of intentionality," *The Cambridge Companion to Brentano*, pp. 98–130, 2004.

[10] F. Dretske, "The explanatory role of content," 1988.

[11] J. Conklin, "Dialog mapping: Reflections on an industrial strength case study," in *Visualizing argumentation*. Springer, 2003, pp. 117–136.

[12] V. E. Van Reijswoud, "The structure of business communication: theory, model and application," Ph.D. dissertation, TU Delft, Delft University of Technology, 1996.

[13] F. Dignum and H. Weigand, "Communication and deontic logic," 1995.

[14] H. H. Clark, *Using language*. Cambridge university press, 1996.

[15] H. Weigand, F. Van Der Poll, A. De Moor *et al.*, "Coordination through communication," in *Proc. of the 8th International Working Conference on the Language-Action Perspective on Communication Modelling (LAP 2003)*, 2003, pp. 1–2.

[16] R. Stamper, K. Liu, M. Hafkamp, and Y. Ades, "Understanding the roles of signs and norms in organizations-a semiotic approach to information systems design," *Behaviour & Information Technology*, vol. 19, no. 1, pp. 15–27, 2000.

[17] J. Groenendijk and M. Stokhof, "Dynamic predicate logic," *Linguistics and philosophy*, vol. 14, no. 1, pp. 39–100, 1991.

[18] J. L. Austin, *How to do things with words*. Oxford university press, 1975.

[19] J. R. Searle, *Speech acts: An essay in the philosophy of language*. Cambridge university press, 1969, vol. 626.

[20] D. Vanderveken, "Meaning and speech acts, vol. 2. formal semantics of success and satisfaction," 1991.

[21] H. Bunt, "Interaction management functions and context representation requirements," in *Proceedings of the Twente Workshop on Language Technology: Dialogue Management in Natural Language Systems (TWLT 11)*, 1996, pp. 187–198.

[22] M. Poesio and D. R. Traum, "Conversational actions and discourse situations," *Computational intelligence*, vol. 13, no. 3, pp. 309–347, 1997.

[23] M. Poesio and D. Traum, "Towards an axiomatization of dialogue acts," in *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues (13th Twente Workshop on Language Technology*. Citeseer, 1998.

[24] T. Winograd and F. Flores, *Understanding computers and cognition: A new foundation for design*. Intellect Books, 1986.

[25] E. Bilange, "A task independent oral dialogue model," in *Proceedings of the fifth conference on European chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 1991, pp. 83–88.

[26] J. Carletta, S. Isard, G. Doherty-Sneddon, A. Isard, J. C. Kowtko, and A. H. Anderson, "The reliability of a dialogue structure coding scheme," *Computational linguistics*, vol. 23, no. 1, pp. 13–31, 1997.

[27] H. Alexandersson, H. Tuomenvirta, T. Schmith, and K. Iden, "Trends of storms in nw europe derived from an updated pressure data set," *Climate Research*, vol. 14, no. 1, pp. 71–73, 2000.

[28] C. L. Sidner, "An artificial discourse language for collaborative negotiation," in *AAAI*, vol. 94, 1994, pp. 814–819.

[29] F. Fipa, "specification part 2: Agent communication language," Technical report, FIPA-Foundation for Intelligent Physical Agents, Tech. Rep., 1997.

[30] N. J. Allen and J. P. Meyer, "Affective, continuance, and normative commitment to the organization: An examination of construct validity," *Journal of vocational behavior*, vol. 49, no. 3, pp. 252–276, 1996.

[31] P. Bretier and D. Sadek, "A rational agent as the kernel of a cooperative spoken dialogue system: Implementing a logical theory of interaction," in *International Workshop on Agent Theories, Architectures, and Languages*. Springer, 1996, pp. 189–203.

[32] M. D. Sadek, "Dialogue acts are rational plans," in *The Structure of Multimodal Dialogue; Second VENACO Workshop*, 1991.

[33] D. R. Initiative *et al.*, "Standards for dialogue coding in natural language processing," Report, Tech. Rep., 1997.

[34] J. Allwood, "Dialog as collective thinking," in *New Directions in Cognitive Science. Publications of the Finnish Artificial Intelligence Society. International Conferences*, no. 2, 1997, pp. 222–226.

[35] P. R. Cohen and H. J. Levesque, "Intention is choice with commitment," *Artificial intelligence*, vol. 42, no. 2-3, pp. 213–261, 1990.

[36] D. R. Traum and J. F. Allen, "A" speech acts" approach to grounding in conversation," in *Second International Conference on Spoken Language Processing*, 1992.

[37] D. Traum, J. Bos, R. Cooper, S. Larsson, I. Lewin, C. Matheson, and M. Poesio, "A model of dialogue moves and information state revision," Tech. rept. Deliverable, Tech. Rep., 1999.

[38] J. F. Allen and C. R. Perrault, "Analyzing intention in utterances," *Artificial intelligence*, vol. 15, no. 3, pp. 143–178, 1980.

[39] J. Allwood, "Obligations and options in dialogue," *Think Quarterly*, vol. 3, pp. 9–18, 1994.

[40] T. H. Bui, M. Rajman, and M. Melichar, "Rapid dialogue prototyping methodology," in *International Conference on Text, Speech and Dialogue*. Springer, 2004, pp. 579–586.

[41] J. Allen and M. Core, "Draft of DAMSL: Dialog act markup in several layers," 1997.

[42] D. Jurafsky, E. Shriberg, and D. Biasca, "Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual," *Institute of Cognitive Science Technical Report*, pp. 97–102, 1997.

[43] J. Ang, Y. Liu, and E. Shriberg, "Automatic dialog act segmentation and classification in multiparty meetings," in *Proceedings of the ICASSP*, vol. 1, 2005, pp. I/1061–I/1064.

[44] T. Bub and J. Schwinn, "Verbmobil: The evolution of a complex large speech-to-speech translation system," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 4. IEEE, 1996, pp. 2371–2374.

[45] A. H. Anderson, M. Bader, E. G. Bard, E. Boyle, G. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller *et al.*, "The hcrc map task corpus," *Language and speech*, vol. 34, no. 4, pp. 351–366, 1991.

[46] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[47] G. W. Allport, "Pattern and growth in personality." 1961.

[48] R. Weinberg and D. Gould, "Foundations of exercise and sport psychology," 1999.

[49] H. J. Eysenck, "The science of personality: Nomothetic." 1954.

[50] A. Bandura and R. H. Walters, *Social learning theory*. Prentice-hall Englewood Cliffs, NJ, 1977, vol. 1.

[51] S. Freud, *The ego and the id*. Courier Dover Publications, 2018.

[52] H. J. Eysenck, "The scientific study of personality." 1952.

[53] H. Eysenck, *The biological basis of personality*. Routledge, 2017.

[54] H. J. Eysenck, "Personality, genetics, and behavior: Selected papers," 1982.

[55] H. Eysenck, "Student selection by means of psychological tests—a critical survey," *British Journal of Educational Psychology*, vol. 17, no. 1, pp. 20–39, 1947.

[56] ——, "Personality and experimental psychology." *Bulletin of the British Psychological Society*, 1966.

[57] R. Cattell, "The scientific study of personality," *Harmondsworth: Penguin*, vol. 252, pp. 76–83, 1965.

[58] G. W. Allport, "Personality: A psychological interpretation." 1937.

[59] E. C. Tupes and R. E. Christal, "Recurrent personality factors based on trait ratings," *Journal of personality*, vol. 60, no. 2, pp. 225–251, 1992.

[60] J. M. Digman, "Personality structure: Emergence of the five-factor model," *Annual review of psychology*, vol. 41, no. 1, pp. 417–440, 1990.

[61] L. R. Goldberg, "The structure of phenotypic personality traits." *American psychologist*, vol. 48, no. 1, p. 26, 1993.

[62] T. J. Trull and D. C. Geary, "Comparison of the big-five factor structure across samples of chinese and american adults," *Journal of Personality Assessment*, vol. 69, no. 2, pp. 324–341, 1997.

[63] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *Journal of Machine Learning Research*, vol. 2, no. Feb, pp. 419–444, 2002.

[64] M. H. Bond, H. Nakazato, and D. Shiraishi, "Universality and distinctiveness in dimensions of japanese person perception," *Journal of Cross-Cultural Psychology*, vol. 6, no. 3, pp. 346–357, 1975.

[65] F. M. Cheung, F. J. van de Vijver, and F. T. Leong, "Toward a new approach to the study of personality in culture." *American Psychologist*, vol. 66, no. 7, p. 593, 2011.

[66] J. Oberlander and A. J. Gill, "Individual differences and implicit language: personality, parts-of-speech and pervasiveness," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 26, no. 26, 2004.

[67] F. Mairesse, M. A. Walker, M. R. Mehl, and R. K. Moore, "Using linguistic cues for the automatic recognition of personality in conversation and text," *Journal of artificial intelligence research*, vol. 30, pp. 457–500, 2007.

[68] C. G. DeYoung, "Personality neuroscience and the biology of traits," *Social and Personality Psychology Compass*, vol. 4, no. 12, pp. 1165–1180, 2010.

[69] J. S. Adelstein, Z. Shehzad, M. Mennes, C. G. DeYoung, X.-N. Zuo, C. Kelly, D. S. Margulies, A. Bloomfield, J. R. Gray, F. X. Castellanos *et al.*, "Personality is reflected in the brain's intrinsic functional architecture," *PloS one*, vol. 6, no. 11, p. e27633, 2011.

[70] S. Dhuria, "Natural language processing: An approach to parsing and semantic analysis."

[71] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT press, 1999.

[72] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[73] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[74] M. Hu and B. Liu, "Mining opinion features in customer reviews," in *AAAI*, vol. 4, no. 4, 2004, pp. 755–760.

[75] G. Somprasertsri and P. Lalitrojwong, "Automatic product feature extraction from online product reviews using maximum entropy with lexical and syntactic features," in *Information Reuse and Integration, 2008. IRI 2008. IEEE International Conference on*. IEEE, 2008, pp. 250–255.

[76] N. Kobayashi, K. Inui, and Y. Matsumoto, "Extracting aspect-evaluation and aspect-of relations in opinion mining," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[77] A. Swami, A. Mete, S. Bhosle, N. Nimbalkar, and S. Kale, "Feature extraction and refinement for opinion mining," 2017.

[78] G. Mishne *et al.*, "Experiments with mood classification in blog posts," in *Proceedings of ACM SIGIR 2005 workshop on stylistic analysis of text for information access*, vol. 19, 2005, pp. 321–327.

[79] W. Zhang, H. Xu, and W. Wan, "Weakness finder: Find product weakness from chinese reviews by using aspects based sentiment analysis," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10 283–10 291, 2012.

[80] S. Stymne, "Pre-and postprocessing for statistical machine translation into germanic languages," in *Proceedings of the ACL 2011 Student Session*. Association for Computational Linguistics, 2011, pp. 12–17.

[81] E. Lloret, H. Saggion, and M. Palomar, "Experiments on summary-based opinion classification," in *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*. Association for Computational Linguistics, 2010, pp. 107–115.

[82] C. C. Aggarwal and C. Zhai, *Mining text data*. Springer Science & Business Media, 2012.

[83] W. Wei, H. Liu, J. He, H. Yang, and X. Du, "Extracting feature and opinion words effectively from chinese product reviews," in *Fuzzy Systems and Knowledge Discovery, 2008. FSKD'08. Fifth International Conference on*, vol. 4. IEEE, 2008, pp. 170–174.

[84] R. Dale, H. Moisl, and H. Somers, *Handbook of natural language processing*. CRC Press, 2000.

[85] S. Bird and E. Loper, "Nltk: the natural language toolkit," in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, 2004, p. 31.

[86] D. Ploch, "Exploring entity relations for named entity disambiguation," in *Proceedings of the ACL 2011 Student Session*. Association for Computational Linguistics, 2011, pp. 18–23.

[87] R. R. Larson, "Introduction to information retrieval," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 4, pp. 852–853, 2010.

[88] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of research and development*, vol. 2, no. 2, pp. 159–165, 1958.

[89] J. Savoy, "Ir multilingual resources at unine," *URL: http://members. unine. ch/jacques. savoy/clef/index. html [Stand: 10.04. 2014]*, 2011.

[90] H. Liu and H. Motoda, *Computational methods of feature selection*. CRC Press, 2007.

[91] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1.

[92] L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo, "Supervised feature selection via dependence estimation," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 823–830.

[93] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero-norm with linear models and kernel methods," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1439–1461, 2003.

[94] P. Mitra, C. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 3, pp. 301–312, 2002.

[95] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *Journal of machine learning research*, vol. 5, no. Aug, pp. 845–889, 2004.

[96] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization," *IEEE Transactions on Neural networks*, vol. 21, no. 7, pp. 1033–1047, 2010.

[97] Z. Zhao and H. Liu, "Semi-supervised feature selection via spectral analysis," in *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 2007, pp. 641–646.

[98] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Machine learning*, vol. 53, no. 1-2, pp. 23–69, 2003.

[99] R. O. Duda, P. E. Hart, D. G. Stork *et al.*, *Pattern classification*. Wiley New York, 1973, vol. 2.

[100] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[101] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on knowledge and data engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[102] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.

[103] G. C. Cawley, N. L. Talbot, and M. Girolami, "Sparse multinomial logistic regression via bayesian l1 regularisation," in *Advances in neural information processing systems*, 2007, pp. 209–216.

[104] J. G. Dy and C. E. Brodley, "Feature subset selection and order identification for unsupervised learning," in *ICML*. Citeseer, 2000, pp. 247–254.

[105] S. Alelyani, J. Tang, and H. Liu, "Feature selection for clustering: A review." *Data Clustering: Algorithms and Applications*, vol. 29, pp. 110–121, 2013.

[106] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification and regression," in *Advances in Neural Information Processing Systems*, 1996, pp. 409–415.

[107] C. Domeniconi and D. Gunopulos, "Local feature selection for classification," *Computational Methods of Feature Selection*, p. 211, 2007.

[108] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk e-mail," in *Learning for Text Categorization: Papers from the 1998 workshop*, vol. 62, 1998, pp. 98–105.

[109] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[110] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[111] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[112] Y. Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k2)," in *Soviet Mathematics Doklady*, vol. 27, no. 2, 1983, pp. 372–376.

[113] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[114] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[115] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[116] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986.

[117] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[118] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[119] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[120] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[121] H. Bunt, J. Alexandersson, J.-W. Choe, A. C. Fang, K. Hasida, V. Petukhova, A. Popescu-Belis, and D. R. Traum, "Iso 24617-2: A semantically-based standard for dialogue annotation." in *LREC*. Citeseer, 2012, pp. 430–437.

[122] H. Bunt, "The dit++ taxonomy for functional dialogue markup," in *AAMAS 2009 Workshop, Towards a Standard Markup Language for Embodied Dialogue Acts*, 2009, pp. 13–24.

[123] J. R. Searle, *Speech acts: An essay in the philosophy of language*, 1969, vol. 626.

[124] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational linguistics*, vol. 26, no. 3, pp. 339–373, 2000.

[125] Y. Ji, G. Haffari, and J. Eisenstein, "A latent variable recurrent neural network for discourse relation language models," *arXiv preprint arXiv:1603.01913*, 2016.

[126] R. Fernandez and R. W. Picard, "Dialog act classification from prosodic features using support vector machines," in *Speech Prosody 2002, International Conference*, 2002.

[127] N. Kalchbrenner and P. Blunsom, "Recurrent convolutional neural networks for discourse compositionality," *arXiv preprint arXiv:1306.3584*, 2013.

[128] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," *arXiv preprint arXiv:1603.03827*, 2016.

[129] H. Khanpour, N. Guntakandla, and R. Nielsen, "Dialogue Act Classification in Domain-Independent Conversations Using a Deep Recurrent Neural Network," *Proceedings of COLING*, pp. 2012–2021, 2016.

[130] A. Qadir and E. Riloff, "Classifying sentences as speech acts in message board posts," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 748–758.

[131] D. Jurafsky, R. Bates, N. Coccaro, R. Martin, M. Meteer, K. Ries, E. Shriberg, A. Stolcke, P. Taylor, and C. V. Ess-Dykema, "Automatic detection of discourse structure for speech recognition and understanding," *1997 IEEE Workshop on Speech Recognition and Understanding*, pp. 88—95, 1997.

[132] R. Klaus, C. Noah, S. Elizabeth, B. Rebecca, J. Daniel, T. Paul, M. Rachel, V. E.-D. Carol, V. E.-D. Carol, and M. Marie, "Automatic detection of discourse structure for speech recognition and understanding," *1997 IEEE Workshop on Speech Recognition and Understanding*, pp. 88—95, 1997.

[133] S. N. Kim, L. Cavedon, and T. Baldwin, "Classifying dialogue acts in one-on-one live chats," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 2010, pp. 862–871.

[134] N. Novielli and C. Strapparava, "The role of affect analysis in dialogue act identification," *IEEE Transactions on Affective Computing*, vol. 4, no. 4, pp. 439–451, 2013.

[135] N. Malandrakis, A. Potamianos, E. Iosif, and S. Narayanan, "Distributional semantic models for affective text analysis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 11, pp. 2379–2392, 2013.

[136] S. Kim, D. Luis Fernando, R. E. Banchs, J. Williams, M. Henderson, and K. Yoshino, "Dialog State Tracking Challenge 4: Handbook," 2015.

[137] M. Henderson, B. Thomson, and J. Williams, "The second dialog state tracking challenge," in *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, vol. 263, 2014.

[138] H. Bunt, J. Alexandersson, J.-W. Choe, A. C. Fang, K. Hasida, V. Petukhova, A. Popescu-Belis, and D. R. Traum, "ISO 24617-2: A semantically-based standard for dialogue annotation." in *Proceedings of LREC*, 2012, pp. 430–437.

[139] H. Bunt, J. Alexandersson, J. Carletta, J.-W. Choe, A. C. Fang, K. Hasida, K. Lee, V. Petukhova, A. Popescu-Belis, L. Romary *et al.*, "Towards an ISO standard for dialogue act annotation," in *Proceedings of LREC*, 2010.

[140] H. Bunt, V. Petukhova, D. Traum, and J. Alexandersson, "Dialogue Act Annotation with the ISO 24617-2 Standard," in *Multimodal Interaction with W3C Standards*, 2017, pp. 109–135.

[141] T. Lei, R. Barzilay, and T. Jaakkola, "Molding cnns for text: non-linear, non-consecutive convolutions," *arXiv preprint arXiv:1508.04112*, 2015.

[142] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, 2013, pp. 273–278.

[143] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[144] A. L. Gorin, "Processing of semantic information in fluently spoken language," in *Proceedings of ICSLP*, vol. 2, 1996, pp. 1001–1004.

[145] R. Iyer, M. Ostendorf, and H. Gish, "Using out-of-domain data to improve in-domain language models," *IEEE Signal processing letters*, vol. 4, no. 8, pp. 221–223, 1997.

[146] D. Povey, P. C. Woodland, and M. J. Gales, "Discriminative MAP for acoustic model adaptation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. (ICASSP'03).*, vol. 1, 2003, pp. I–I.

[147] T. Lei, H. Joshi, R. Barzilay, T. Jaakkola, K. Tymoshenko, A. Moschitti, and L. Marquez, "Semi-supervised question retrieval with gated convolutions," *arXiv preprint arXiv:1512.05726*, 2015.

[148] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[149] M. D. Ekstrand, J. T. Riedl, J. A. Konstan *et al.*, "Collaborative filtering recommender systems," *Foundations and Trends® in Human–Computer Interaction*, vol. 4, no. 2, pp. 81–173, 2011.

[150] R. Gao, B. Hao, S. Bai, L. Li, A. Li, and T. Zhu, "Improving user profile with personality traits predicted from social media content," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 355–358.

[151] A. Aly and A. Tapus, "Towards an intelligent system for generating an adapted verbal and nonverbal combined behavior in human–robot interaction," *Autonomous Robots*, vol. 40, no. 2, pp. 193–209, 2016.

[152] P. Fung, A. Dey, F. B. Siddique, R. Lin, Y. Yang, Y. Wan, and H. Y. R. Chan, "Zara the supergirl: An empathetic personality recognition system," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 2016, pp. 87–91.

[153] C. S. Carver, S. K. Sutton, and M. F. Scheier, "Action, emotion, and personality: Emerging conceptual integration," *Personality and social psychology bulletin*, vol. 26, no. 6, pp. 741–751, 2000.

[154] R. M. Ryan, "A motivational approach to self: Integration in personality edward l., deci and," *Perspectives on motivation*, vol. 38, no. 237, pp. 237–288, 1991.

[155] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," *CoRR*, vol. abs/1403.6382, 2014.

[156] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014.

[157] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 1615–1625. [Online]. Available: https://www.aclweb.org/anthology/D17-1169

[158] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," *ArXiv e-prints*, Jan. 2018.

[159] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[160] D. Xue, Z. Hong, S. Guo, L. Gao, L. Wu, J. Zheng, and N. Zhao, "Personality recognition on social media with label distribution learning," *IEEE Access*, vol. 5, pp. 13 478–13 488, 2017.

[161] J.-I. Biel and D. Gatica-Perez, "The youtube lens: Crowdsourced personality impressions and audiovisual analysis of vlogs," *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 41–55, 2013.

[162] J. W. Pennebaker and L. A. King, "Linguistic styles: Language use as an individual difference." *Journal of personality and social psychology*, vol. 77, no. 6, p. 1296, 1999.

[163] S. Poria, A. Gelbukh, B. Agarwal, E. Cambria, and N. Howard, "Common sense knowledge based personality recognition from text," in *Mexican International Conference on Artificial Intelligence*. Springer, 2013, pp. 484–496.

[164] M. P. Kalghatgi, M. Ramannavar, and D. N. S. Sidnal, "Social-network-sourced big data analytics for personality prediction: A review," 2015.

[165] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: Liwc 2001," *Mahway: Lawrence Erlbaum Associates*, vol. 71, no. 2001, p. 2001, 2001.

[166] S. M. Mohammad and S. Kiritchenko, "Using hashtags to capture fine emotion categories from tweets," *Computational Intelligence*, vol. 31, no. 2, pp. 301–326, 2015.

[167] F. Liu, J. Perez, and S. Nowson, "A language-independent and compositional model for personality trait recognition from short texts," *arXiv preprint arXiv:1610.04345*, 2016.

[168] S. Gievska and K. Koroveshovski, "The impact of affective verbal content on predicting personality impressions in youtube videos," in *Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition*. ACM, 2014, pp. 19–22.

[169] M. Coltheart, "The mrc psycholinguistic database," *The Quarterly Journal of Experimental Psychology Section A*, vol. 33, no. 4, pp. 497–505, 1981.

[170] M. Wilson, "Mrc psycholinguistic database: Machine-usable dictionary, version 2.00," *Behavior research methods, instruments, & computers*, vol. 20, no. 1, pp. 6–10, 1988.

[171] J. Yu and K. Markov, "Deep learning based personality recognition from facebook status updates," in *Awareness Science and Technology (iCAST), 2017 IEEE 8th International Conference on*. IEEE, 2017, pp. 383–387.

[172] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," 09 2014.

[173] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Parikh, and D. Batra, "Vqa: Visual question answering," *Int. J. Comput. Vision*, vol. 123, no. 1, pp. 4–31, May 2017.

[174] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: a large-scale hierarchical image database," pp. 248–255, 06 2009.

[175] C. Baziotis, N. Athanasiou, A. Chronopoulou, A. Kolovou, G. Paraskevopoulos, N. Ellinas, S. Narayanan, and A. Potamianos, "Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning," *arXiv preprint arXiv:1804.06658*, 2018.

[176] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[177] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, “Hypercolumns for object segmentation and fine-grained localization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 447–456.

[178] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, “Dailydialog: A manually labelled multi-turn dialogue dataset,” *arXiv preprint arXiv:1710.03957*, 2017.

[179] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting.” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[180] Y. R. Tausczik and J. W. Pennebaker, “The psychological meaning of words: Liwc and computerized text analysis methods,” *Journal of language and social psychology*, vol. 29, no. 1, pp. 24–54, 2010.

Appendix A

Abbreviations

(DA): Dialogue Act
(SDS): Spoken Dialogue System
(DNN): Deep Neural Network
(LSTM): Long Short-Term Memory
(CNN): Convolutional Neural Network
(AI): Artificial Intelligence
(DS): Dialogue System
(NLP): Natural Language Processing
(ML): Machine Learning
(DL): Deep Learning
(BiLSTM): Bidirectional LSTM
(TL): Transfer Learning
(LAP): Language/Action Perspective
(TPM): Transaction Process Model
(CA): Communicative Act
(RPDM): Rapid Dialogue Prototype Methodology
(ASR): Automatic Speech Recognition
(DAMSL): Dialogue Act Markup in Several Layers
(MRDA): Meeting Recorder DA
(ANS): Automatic Nervous System
(IR): Information Retrieval
(MT): Machine Translation
(IE): Information Extraction
(QA): Question Answering
(POS): Part-Of-Speech
(kNN): k-Nearest Neighbor
(MAP): Maximum A Posteriori
(SVM): Support Vector Machine
(CBOW): Continuous Bag-Of-Words
(ANN): Artificial Neural Network
(MLP): Multi-Layer Perceptron
(FFNN): Feed-Forward Neural Network
(ReLU): Rectified Linear Unit
(GD): Gradient Descent
(SGD): Stochastic Gradient Descent
(RNN): Recurrent Neural Network
(BiRNN): Bidirectional Recurrent Neural Network
(CV): Computer Vision
(INIT): Parameter Initialization
(MULT): MUlti-Task learning
(HMM): Hidden Markov Models
(HAN): Hierarchical Attention Network

(FFM): Five Factor Model
(LIWC): Linguistic Inquiry Word Count
(LM): Language Model
(SE): Sentence Encoder
(INIT-TL): Initialization of sentence encoder
(HTL): Hypercolumns transfer learning of sentence encoder
(MSE): Mean Square Error
(RMSE): Root Mean Square Error