Feature Expansion for Social Media User Characterization

André Filipe da Cruz Monteiro

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
(2º ciclo de estudos)

Orientador: Prof. Doutor João Paulo da Costa Cordeiro

março de 2021

Agradecimentos

Em primeiro lugar, agradeço ao meu orientador, Professor Doutor João Paulo Cordeiro, por toda a disponibilidade, orientação e ajuda.

Agradeço aos meus pais e irmã por me terem guiado e acompanhado no caminho certo do meu desenvolvimento pessoal, educacional e profissional.

À Diana, pelo seu apoio e carinho incondicionais, por me ter motivado a dar o meu melhor.

Por fim, aos meus amigos com quem partilhei esta jornada.

A todos, o meu mais sincero obrigado.

Resumo

A personalidade é um fator fundamental nas nossas vidas e os psicólogos acreditam que o comportamento de um indivíduo pode ser inferido com base na sua personalidade. Recentemente, ocorreram casos de disseminação de informação falsa em redes sociais por parte de pessoas influentes, executando assim ações potencialmente perigosas. Para prevenir estes acontecimentos, é necessário identificar quais os utilizadores que afetarão negativamente a comunidade, e poderemos fazê-lo com o reconhecimento de personalidade através das suas publicações em redes sociais.

Esta dissertação apresenta uma abordagem à tarefa de reconhecimento de personalidade através de texto. Durante a revisão bibliográfica, identificámos uma ferramenta de análise de texto chamada Linguistic Inquiry and Word Count (LIWC) que é usada repetidamente e com sucesso em trabalhos relacionados e, portanto, decidimos que será a base de dados a utilizar para extração de características. Verificou-se também que classificadores Support-Vector Machine produzem os melhores resultados. Perante estes factos, delineámos os seguintes objetivos: (i) explorar algoritmos de aprendizagem automática diferentes dos usados em trabalhos relacionados para encontrar um que produza melhores resultados; (ii) analisar se uma extensão não supervisionada do vocabulário do LIWC melhora os resultados da classificação.

Para treinar e testar os modelos, usámos um conjunto de 2468 ensaios de fluxo de consciência anotados com os traços de personalidade Big Five do escritor: abertura para a experiência, conscienciosidade, extroversão, amabilidade, e neuroticismo. Implementámos quatro algoritmos de aprendizagem automática para classificar os textos: *Support-Vector Machine*, *Naive Bayes*, *Decision Tree*, e *Random Forest*. Para além disso, selecionámos dois métodos para a expansão de vocabulário: sinónimos cognitivos do *WordNet*, e *Word Embeddings*.

Os resultados obtidos demonstram que o classificador Random Forest tem uma performance promissora, semelhante à dos algoritmos utilizados pelos artigos relacionados, com uma exatidão média de aproximadamente 56.5%. As expansões de vocabulário realizadas traduziram-se num aumento de 0.6% de palavras dos ensaios atribuídas a categorias do LIWC. No entanto, a diferença introduzida nos resultados não é significativa, portanto a expansão de vocabulário não mostrou benefícios.

Palavras-chave

Aprendizagem Automática, Big Five, Classificação, Identificação de Personalidade, Inteligência Artificial, Processamento de Linguagem Natural, Psicologia da Personalidade

Resumo alargado

A personalidade é um fator fundamental nas nossas vidas, como nas nossas relações interpessoais, no nosso desempenho profissional, e na forma como escrevemos. A expressão escrita de um indivíduo transporta informação sobre as suas crenças, medos, formas de pensar, relações sociais, e personalidade [1]. Os psicólogos acreditam que o comportamento de um indivíduo pode ser inferido com base na sua personalidade.

Recentemente, ocorreram casos de disseminação de informação falsa em redes sociais por parte de pessoas influentes, executando assim ações potencialmente perigosas. Um exemplo de acontecimentos deste tipo é a divulgação de notícias e artigos antivacinação, os quais contribuem para a desinformação e, como consequência, para um aumento no número de doentes. De modo a prevenir estes acontecimentos, é necessário identificar quais os utilizadores que afetarão negativamente a comunidade, e poderemos fazê-lo com o reconhecimento de personalidade através das suas publicações em redes sociais.

Durante a revisão bibliográfica, identificámos uma ferramenta de análise de texto chamada LIWC que é usada repetidamente e com sucesso em trabalhos relacionados. Esta ferramenta encontra-se dividida em dois componentes: um módulo de processamento de texto, e um dicionário de suporte. O dicionário de suporte contém 6548 palavras e radicais associados a 73 categorias, e é a base escolhida para extração de características.

Verificámos também que classificadores Support-Vector Machine produzem os melhores resultados. Perante estes factos, delineámos os seguintes objetivos: (i) explorar algoritmos de aprendizagem automática diferentes dos usados em trabalhos relacionados para encontrar um que produza melhores resultados; (ii) analisar se uma extensão não supervisionada do vocabulário do LIWC melhora os resultados da classificação.

O nosso conjunto de dados de entrada consiste em 2468 ensaios de fluxo de consciência anotados com os traços de personalidade Big Five do escritor - abertura para a experiência, conscienciosidade, extroversão, amabilidade, e neuroticismo - reunidos pelos autores do LIWC e redigidos por alunos de Pennebaker, aos quais foi pedido que escrevessem continuamente, por 20 minutos, sobre o que pensam e sentem acerca de terem iniciado o ensino superior.

Durante este trabalho, desenvolvemos um algoritmo em Python que recorre às bibliotecas Natural Language Toolkit (NLTK) e Scikit-learn para as tarefas de pré-processamento, classificação, expansão de vocabulário, e posterior avaliação dos resultados. Foram implementados quatro algoritmos de aprendizagem automática para classificar os textos: *Support-Vector Machine*, *Naive Bayes*, *Decision Tree*, e *Random Forest*. Para além disso, selecionámos dois métodos para a expansão de vocabulário: sinónimos cognitivos do

WordNet, e *Word Embeddings*.

Durante o pré-processamento, cada ensaio foi analisado para criar dados de saída semelhantes aos do módulo de processamento do LIWC: número total de palavras do texto, número de palavras por frase, percentagem de palavras de cada categoria, percentagem de palavras com mais de seis letras, e percentagem de palavras encontradas no dicionário de suporte. Neste conjunto não se encontram incluídas quatro variáveis de resumo (pensamento analítico, influência, autenticidade, e tom emocional) pois os algoritmos são proprietários e derivam de pesquisas anteriores dos autores da ferramenta.

A expansão de vocabulário com recurso ao WordNet consistiu em encontrar os sinónimos cognitivos mais próximos de cada palavra do dicionário do LIWC. A extensão com recurso a Word Embeddings, nomeadamente à base de dados word2vec, traduziu-se na procura de palavras semelhantes através de operações aritméticas: como as palavras se encontram sob a forma de vetores numéricos, foi possível utilizar a similaridade cosseno para calcular a similaridade entre duas palavras.

Para dividir os dados de treino e de teste e validar os modelos criados utilizámos o método de validação cruzada *k-fold* com $k = 10$. Para cada classificador foram usados os hiperparâmetros predefinidos no Scikit-learn. Para comparação e discussão de resultados, recolhemos quatro diferentes valores estatísticos: exatidão (*accuracy*), precisão (*precision*), sensibilidade (*recall*), e *F1-score*.

Os resultados obtidos demonstram que o classificador Random Forest tem uma performance promissora, semelhante à dos algoritmos utilizados pelos artigos relacionados, com uma exatidão média de aproximadamente 56.5%. As expansões de vocabulário realizadas traduziram-se num aumento de 0.6% no total de palavras dos ensaios atribuídas a categorias do LIWC. No entanto, a diferença introduzida nos resultados não é significativa, portanto a expansão de vocabulário não mostrou benefícios.

Abstract

Personality plays an impactful role in our lives and psychologists believe that an individual's behavior can be inferred through its personality. Recently, there have been cases of influential people in social media spreading misinformation, which is a potentially dangerous action. To prevent it, we need to identify which users will negatively impact the community, and we might be able to predict such behavior through personality recognition from their social media posts.

This dissertation presents an approach to personality recognition from text. During the bibliographic revision, we learned that a text analysis tool called LIWC is repeatedly used with success for tasks of this type, thus we chose the LIWC dictionary to be the base feature set to consider. Also, we have found that Support-Vector Machine classifiers exhibit the best results. From these two findings, we outlined the following objectives: (i) exploit machine learning algorithms different from the ones used in related works to find one that produces better results; (ii) analyze whether extending LIWC's vocabulary without supervision improves the classification results.

For training and testing, we used a data set of stream-of-consciousness essays comprised of 2468 samples annotated with the Big Five personality traits of the writer: openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism. We used four machine learning algorithms for classification: Support-Vector Machine, Naive Bayes, Decision Tree, and Random Forest. Also, we selected two methods for vocabulary expansion: WordNet's synsets, and Word Embeddings.

The results obtained show that the Random Forest classifier performs similarly to the algorithms used in related works, with an average accuracy of approximately 56.5%, which are promising ratings. The vocabulary expansions we have performed allowed the algorithm to match 0.6% more words from the essay data set. However, the changes to the classification results were not significant, therefore the vocabulary expansion was not beneficial.

Keywords

Artificial Intelligence, Big Five, Classification, Machine Learning, Natural Language Processing, Personality Psychology, Personality Recognition

Contents

List of Figures

List of Tables

List of Acronyms

16PF	Sixteen Personality Factors
KDD	Knowledge discovery in databases
LIWC	Linguistic Inquiry and Word Count
MRC	Medical Research Council
NEO-PI	NEO Personality Inventory
NEO-PI-R	Revised NEO Personality Inventory
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NRC	National Research Council
SMO	Sequential Minimal Optimization
SVM	Support-Vector Machine
WN	WordNet
WNA	WordNet-Affect

Chapter 1

Introduction

This document details the work developed in the scope of the project for obtaining a Master's degree in Computer Science and Engineering at Universidade da Beira Interior. This dissertation addresses the task of recognizing an individual's personality from texts written by them and assesses whether existing works in the field can be improved by a different classification approach.

1.1 Motivation and Scope

Personality plays an impactful role in our lives - for instance, on inter-personal relationships and job performance [12]. Raymond Cattell, one of the most productive psychologists of the 20th century, defined personality as "that which permits a prediction of what a person will do in a given situation" [13], which means, in other words, that we may be able to infer the behavior of an individual if we identify their personality.

Previous works found out that the utterances convey rich information about people's beliefs, fears, thinking patterns, social relationships, and personalities [1]. As such, it is only natural to think that we may be able to extract knowledge about a person from written text authored by them.

In the latest years, the crescent number of social media users lead them to have a preponderant role in public opinion formation and awareness. To give a perspective on the user base growth, according to [14] Facebook has grown from approximately 518 million users in 2010 to 2.38 billion in 2019. With this tendency, users with an enormous number of followers have gained the power to promote products and ideas to a large audience, thus started being called influencers.

This influential power has proven to be nocive in certain cases. For instance, during the Covid-19 pandemic of 2020, there were influencers promoting widely-spread fake news and articles against vaccination and the use of masks, which furthered misinformation and, consequently, the spread of the disease.

To prevent this injurious behavior, we need to know which users will negatively impact the community. We might be able to perceive what course of action a user will take for a

certain situation if we identify beforehand their personality traits and, if the possibility of personality recognition from text is real, we might be able to infer these personality traits from the influencers' social media posts from, for instance, Twitter or Facebook.

1.2 Problem Statement and Objectives

The reviewed literature has shown us that the use of LIWC dictionary shows promising results for personality recognition from text, and as such it will be the base for this Masters project.

During the literature review we have found two points on which we can improve. First, we found a lack of experimentation with classifiers other than support-vector machines. Second, we found that the authors extract features not only from LIWC but also from other databases and lexicons, such as the MRC Psycholinguistic Database and EmoSenticNet, thus creating an opportunity to experiment only with LIWC.

In order to tackle these gaps, we defined the following set of objectives:

- Explore a variety of machine learning algorithms for classification, adequate to Natural Language Processing (NLP) tasks;
- Expand LIWC's vocabulary through unsupervised methods in order to increase the range of words it can match.

The ultimate goal is to assess whether different classification algorithms produce better results than the ones reported in related works and to evaluate the benefits of an expanded original data set on personality identification from text.

1.3 Approach

After defining the problems we are going to tackle, as well as the objectives we aim to achieve by addressing these gaps, we outline the approach we will take.

Our approach begins by choosing a set of machine learning algorithms for classification that are adequate for the task at hand and selecting the databases to consult in order to expand the initial vocabulary without supervision.

After selecting the expansion databases, we will create additional data sets for each possible combination of expansion methods. For illustration purposes, assume D is the default

data set, and X and Y are the extensions we have created for two different databases; in this case, there are four possible combinations of data: D, DX, DY, and DXY. Our subsequent steps will be performed on each of those combinations.

Then, we preprocess the input data, and train and test the classifiers using the results of the preprocessing task. This step will produce statistics from the outputs of the classifiers.

After gathering metrics for each possible dictionary, for each classifier, we will compare the results in two different ways:

1. Analyze the difference between the values obtained using the default dictionary with the results obtained by the enlargened dictionaries;
2. Compare the results obtained in our experiments with the ratings reported in related works.

To meet these ends, we developed an algorithm that uses the following collection of data sets, lexicons, and technologies:

- **Python** - the programming language in which the algorithm was developed;
- **NLTK** - library with features to work with human language;
- **Scikit-learn** - machine learning module for Python that we use to create, train and test, and evaluate classifiers;
- **LIWC dictionary** - maps words to categories, it is the default data set we used to extract features from the input data;
- **WordNet** - a database that links words with cognitive synonyms, we used it to find synonyms of the original vocabulary;
- **word2vec** - database with word embeddings, used to search for words that are similar to the ones in the original vocabulary;

1.4 Main Contributions

The main contributions this work may have can be enumerated as follows:

1. Study the state of the art on personality recognition from text so that we have a better understanding of what has been done in this specific field.

2. Exploitation of a new machine learning algorithm that produces better results for the personality classification task than the ones reported in similar works;
3. Analysis of the LIWC dictionary and assessment of whether extending it is beneficial for the task of personality identification from text;

1.5 Document Overview

The organization of this document is summarized as listed below:

1. Introduction — presents the current project's motivation and scope, the problems it attempts to solve, the approach the authors take to solve the identified problems, the main contributions of this work, and the organization of this document;
2. Background and Related Works - introduces the concepts that serve as the cornerstone for this work; explains the psychological foundations for the project, the knowledge discovery in databases process, the natural language processing techniques and lexicons used, the machine learning algorithms and tools implemented and reviews the existing literature on this work's topics.
3. Methodology - clarifies the practical component of this project, namely the preprocessing of the data, the vocabulary expansion, and the classification steps;
4. Results and Discussion - elaborates on the results obtained by the classification step, and compares them to each other and to related works;
5. Conclusions and Future Work - presents the final comments of this investigation, what was achieved, and future work can originate from it.
6. Appendix A - Demonstrations - details lengthier calculations as a complement to the information described in chapter 2.
7. Appendix B - Data samples - shows integral samples of the data set used in this work.

Chapter 2

Background and Related Works

2.1 Introduction

The present chapter first introduces the psychology foundations for this work in Section 2.2. In Section 2.3 we enumerate all the steps involved in the KDD process. In Section 2.4 we establish some fundamentals of linguistics and explore various natural language tools and lexicons, including NLTK. Section 2.5 presents four different machine learning algorithms that will be used for classification and takes a tour through the machine learning library Scikit-learn. Finally, we review the existing literature on this work's topics in Section 2.6.

2.2 Psychology Foundations

This section presents the state of the art definitions of personality in Section 2.2.1. Then, in Sections 2.2.2 we explain in more detail the trait approach to personality is detailed, and briefly introduce existing personality trait models. In the end, we define the link between personality and emotion in Section 2.2.3.

2.2.1 Personality

Personality is considered a fragmented subject, as scientists have "not agreed upon a single shared paradigm that would foster cooperation and steady incremental scientific growth" [9]. Among the several existing perspectives on psychology, the most known and developed are the biological perspective, the cognitive perspective, the humanistic perspective, the learning perspective, the psychodynamic perspective, and the trait perspective. Table 2.1 introduces the major concepts for each of these perspectives, and their largest contributors. This thesis focuses on the trait perspective, which is described in detail in Section 2.2.2.

Definitions of personality vary from perspective to perspective. Raymond Cattell uses traits to predict behavior, defining personality as "that which permits a prediction of what

a person will do in a given situation" [13]. Gordon Allport describes personality as "the dynamic organization within the individual of those psychophysical systems that determine his unique adjustments to the environment", highlighting the integration of personality, i.e. the assimilation of new experiences, data, and emotional competencies into the personality [15]. McAdams and Pals, on a more sophisticated take, characterize personality as "an individual's unique variation on the general evolutionary design for human nature, expressed as a developing pattern of dispositional traits, characteristic adaptations, and integrative life stories complexly and differentially situated in culture" [16].

Table 2.1: Major perspectives in personality, as reported in [9].

Perspective	**Major concepts**	**Contributors**
Biological	temperament, evolution, adaptation, altruism, sexual jealousy, heredity, neurotransmitter pathways, cerebral hemisphere function	D. Buss, Eysenck, J. A. Gray, C. R. Cloninger, Kagan
Cognitive	expectancy, self-efficacy, outcome expectation, schema, cognitive person variable, personal construct, reciprocal determinism, modelling, constructive alternativism, life narrative	Mischel, Bandura, Kelly, Beck
Humanistic	self-actualization, creativity, flow, spirituality, personal responsibility, freedom, choice, openness to experience, unconditional positive regard, acceptance, empathy, real self, hierarchy of needs, peak experience, positive psychology	Maslow, Rogers, Seligman, Csikszentmihalyi
Learning	reinforcement, punishment, stimulus, response, conditioning, extinction, shaping, discrimination learning, generalization, situation, act frequency, basic behavioural repertoire, labelling, gradients of approach and avoidance	Skinner, Staats, Dollard and Miller
Psychodynamic	libido, conflict, id, ego, superego, defence mechanisms, Oedipal conflict, fixation, repression, attachment, object-relations	Freud, Jung, Adler, Erikson, Horney, Klein, Sullivan, Chodorow, Westen, Kohut, Kernberg
Trait	trait, type, facet, factors, Neuroticism/ Emotional Stability, Extraversion	Allport, Cattell, McCrae and Costa

2.2.2 The trait approach to personality

Personality psychology is concerned with the dispositions in which individuals from the same age range differ from each other. Dispositions are temporally stable tendencies of behaviur, more simplified, a tendency for someone to act in a certain way. The personality of an individual can be defined by dispositions, and those dispositions are called personality dispositions, or personality traits [9].

Researchers have been trying, for more than a century, to find what the nature of personality traits is, and how to identify them. Allport created a doctrine to explain what a trait of personality is [17]:

1. A trait has more than nominal existence - a trait is something that in fact exists, the same way a habit exists.
2. A trait is more generalized than a habit - traits involve habits that are related to each other either statistically or genetically; e.g., a trait can derive from the relation between the habit of stealing and the habit of brushing teeth only if it is possible to prove that the two are related.
3. A trait is dynamic, or at least determinative - traits are considered drives that arise from specific stimuli, specifying in the process the way one reacts to the stimuli.
4. The existence of a trait may be established empirically or statistically - to assert that a person has a habit, evidence of "repeated reactions of a constant type" is needed. In the same way, to know that a person has a trait, it is necessary to have evidence of repeated, consistent reactions.
5. Traits are only relatively independent of each other - studies show that when studying traits through observation of concrete acts, those acts reflect not only the trait under examination but also other traits simultaneously.
6. A trait of personality, psychologically considered, is not the same as moral quality - traits' names should be, as much as possible, devalued of any moral significance, as it can lead to limiting its concept by conventional meanings.
7. Acts, and even habits, that are inconsistent with a trait are not proof of the non-existence of the trait - consider the following sentence: "an individual may be habitually neat with respect to his person, and characteristically slovenly in his handwriting or the care of his desk". First, a trait can be minor, major, or even non-existing, depending on the degree of integration on a person, i.e. there is evidence that, for instance, the trait of neatness is integrated into every act of a person, but not all people show that trait. Second, a person can have contradictory traits, e.g., both neatness and carelessness. Third, there are acts that do not reflect existing traits that are the product of the circumstances.
8. A trait may be viewed either in the light of the personality which contains it or in the light of its distribution in the population at large - each trait has two aspects: the unique aspect and the universal aspect. The unique aspect is concerned about the role it plays in the personality as a whole, while the universal aspect is the separate study of the trait and the comparison, in respect to it, of individuals of a determined population to find individual differences.

In 1943, Raymond Cattell introduced the concept of "personality sphere", which he described as "the full description of personality defining the factor space for basic personality variables" [18]. Cattell condensed a list of 171 trait verbal definitions, which later reduced to a set of 35 traits clustered from correlations found from the ratings of 100 individuals. This group of personality traits is considered the common origin of three of the most popular personality trait models: the sixteen primary factor model, the Big Five model, and the NEO three factor model [9]. These models are described in more detail in the next sections.

2.2.2.1 The sixteen-factor model

The Sixteen Personality Factors (16PF) model is based on the 35 traits described in 2.2.2. Cattell created the 16PF model through a centroid analysis of the 35 trait ratings of 208 adult men. The investigation showed that the 35 traits could be reduced to twelve; Cattell added four "questionnaire-specific scales" to these twelve factors, totaling 16 personality factors. [19][9].

In 1995, the fifth edition of the 16PF was published [20]. Table 2.2 shows the 16 personality factors, alongside a description of the lower and higher ranges of each trait.

2.2.2.2 The NEO-PI model

Costa and McCrae began the creation of the NEO Personality Inventory (NEO-PI) three-factor model during the investigation of whether there are structural changes to the personality with age. To first determine what the personality structure involves, the authors summarized the 16PF model as three factors: neuroticism (N), extraversion (E), and openness (O). Each of these domains is described by a set of specific traits named facets [10].

A revised five-factor model - Revised NEO Personality Inventory (NEO-PI-R) - was published in 1992, with the aim of solving the major limitation of NEO-PI: the lack of facet scales for agreeableness (A) and conscientiousness (C), which were present in the work of other researchers and were proved to strengthen them [10].

Each dimension of NEO-PI-R has six facet scales [21]:

- Neuroticism:

  N1: Anxiety. Calm, relaxed vs. tense, worried.

  N2: Angry Hostility. Even-tempered, gentle vs. hot-tempered, frustrated.

Table 2.2: List of Catell's 16 primary personality factors, as in [10]

Descriptors of low range	**Primary factor**	**Descriptors of high range**
Reserved, Impersonal, Distant	Warmth (A)	Warm-hearted, Caring, Attentive To Others
Concrete, Lower Mental Capacity	Reasoning (B)	Abstract, Bright, Fast-Learner
Reactive, Affected By Feelings	Emotional Stability (C)	Emotionally Stable, Adaptive, Mature
Deferential, Cooperative, Avoids Conflict	Dominance (E)	Dominant, Forceful, Assertive
Serious, Restrained, Careful	Liveliness (F)	Enthusiastic, Animated, Spontaneous
Expedient, Nonconforming	Rule-Consciousness (G)	Rule-Conscious, Dutiful
Shy, Timid, Threat-Sensitive	Social Boldness (H)	Socially Bold, Venturesome, Thick-Skinned
Tough, Objective, Unsentimental	Sensitivity (I)	Sensitive, Aesthetic, Tender-Minded
Trusting, Unsuspecting, Accepting	Vigilance (L)	Vigilant, Suspicious, Skeptical, Wary
Practical, Grounded, Down-To-Earth	Abstractedness (M)	Abstracted, Imaginative, Idea-Oriented
Forthright, Genuine, Artless	Privateness (N)	Private, Discreet, Non-Disclosing
Self-Assured, Unworried, Complacent	Apprehension (O)	Apprehensive, Self-Doubting, Worried
Traditional, Attached To Familiar	Openness to Change (Q1)	Open To Change, Experimenting
Group-Orientated, Affiliative	Self-Reliance (Q2)	Self-Reliant, Solitary, Individualistic
Tolerates Disorder, Unexacting, Flexible	Perfectionism (Q3)	Perfectionistic, Organized, Self-Disciplined
Relaxed, Placid, Patient	Tension (Q4)	Tense, High Energy, Driven

N3: Depression. Hopeless, guilty, downhearted vs. hopeful, confident, feeling worthwhile.

N4: Self-Consciousness. Secure, feeling adequate vs. ashamed, feeling inferior.

N5: Impulsiveness. Self-controlled vs hasty, unable to resist cravings.

N6: Vulnerability. Resilient, cool-headed vs panicked, unable to deal with stress.

- Extraversion:

E1: Warmth. Formal, reserved vs. outgoing, affectionate.

E2: Gregariousness. Preferring to be alone vs. seeking social contact.

E3: Assertiveness. Reticent, unassuming vs. dominant, forceful.

E4: Activity. Unhurried, deliberate vs. energetic, fast-paced.

E5: Excitement-Seeking. Cautious vs. thrill-seekers.

E6: Positive Emotions. Serious, unenthusiastic vs. cheerful, joyful, optimistic.

- Openness To Experience:

  O1: Fantasy. Practical, realistic vs. imaginative.

  O2: Aesthetics. Insensitive to art vs. valuing art and beauty.

  O3: Feelings. Insensitive to surrounding emotions, do not believe that feelings are of much importance vs. empathic, sensitive, values own feelings.

  O4: Actions. Preferring the comfort of the familiar, avoiding changes vs. avoiding routine, trying new activities.

  O5: Ideas. Pragmatic, limited curiosity vs. intellectually curious.

  O6: Values. Conservative, conforming vs. tolerant, nonconforming, open-minded.

- Agreeableness:

  A1: Trust. Skeptical, pessimistic, suspicious vs. being forgiving, trusting.

  A2: Straightforwardness. Astute, clever, charming vs. direct, frank, naive.

  A3: Altruism. Selfish, cynical, snobbish vs. warm, generous, kind.

  A4: Compliance. Stubborn, demanding vs. respectful, helpful.

  A5: Modesty. Aggressive, tending to show off vs. humble.

  A6: Tender-Mindedness. Intolerant, cold vs. friendly, soft-hearted, moved by other's needs.

- Conscientiousness:

  C1: Competence. Confused, forgetful, carefree vs. efficient, prudent, confident, intelligent.

  C2: Order. Disorderly, untidy, impulsive vs. precise, methodical.

  C3: Dutifulness. Lazy, distractible vs. dependable, organized.

  C4: Achievement Striving. Relaxed, disorganized, lacking ambition vs. ambitious, persistent.

  C5: Self-Discipline. Unambitious, forgetful vs. motivated, energetic, capable, efficient.

  C6: Deliberation. Hasty, immature, careless vs. cautious, logical, mature.

2.2.2.3 The Big Five model

Tupes and Christal, in 1961, found five strong, recurring factors from the group of 35 trait variables, and labeled them surgency, agreeableness, dependability, emotional stability, and culture. The authors considered these factors to be fundamental and universal, but not the only personality dimensions that may exist [22]. In [23], Norman, in his work

about the five factors taxonomy, renamed the dependability trait to conscientiousness, and referred to the surgency trait also as extroversion.

In the decade of 1980, investigators such as John M. Digman and Lewis Goldberg started reviewing the existing personality instruments, which led the personality researchers to widely accept the five-factor [24][25]. The term "Big Five" was coined by Goldberg, in 1981.

Big Five, therefore, refers to a generic structure accepted by the researcher community and, as such, there are several instruments to assess one's Big Five. One example of an instrument is the NEO-PI questionnaire.

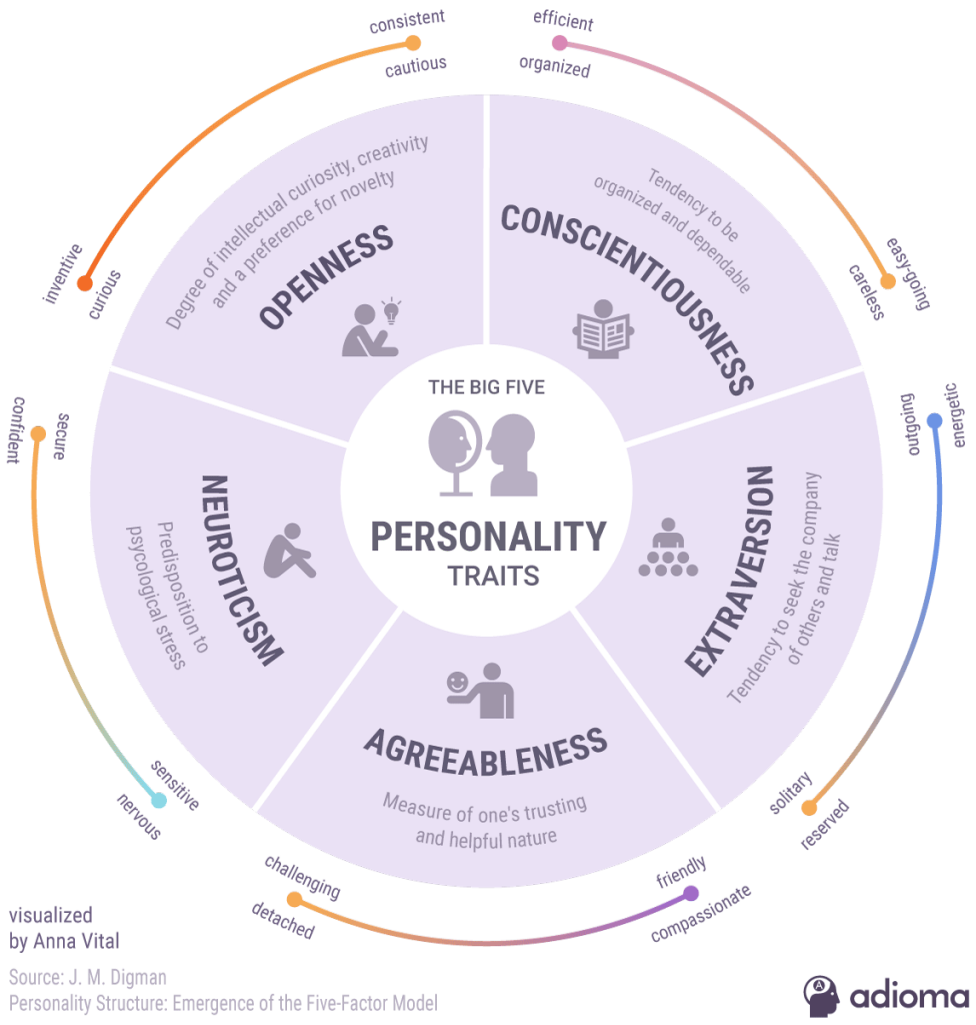


Figure 2.1: Big Five infographic sourced from [2].

2.2.3 Personality and Emotion

There is no general consensus for a definition of emotion. However, there are two points that the researchers agree on. First, the research on emotions focuses on "the transi-

tory states of persons denoted by ordinary language words such as 'happiness', 'sadness', 'fear', 'anger', 'pity', 'pride', 'guilt', and so forth". Second, emotional experiences occur as reactions to the perception or imagination of "objects" (specific events or situations) and can have both subjective and objective manifestations. Emotions manifest subjectively as pleasant or unpleasant feelings directed at a trigger object, for instance, feeling happy or mad about the arrival of a person, whether they like the person or not. Objectively, emotions manifest in particular actions (e.g., flight in case of fear), expressive reactions (e.g., smiling when happy), and physiological changes (e.g., angry due to rise in blood pressure) [9].

Plutchik created a set of eight basic emotions - acceptance, anger, anticipation, disgust, joy, fear, sadness, and surprise - based on his own theory of emotions' ten postulates [26]:

1. The concept of emotion is applicable to all evolutionary levels and applies to animals as well as to humans.
2. Emotions have an evolutionary history and have evolved various forms of expression in different species.
3. Emotions serve an adaptive role in helping organisms deal with key survival issues posed by the environment.
4. Despite different forms of expression of emotions in different species, there are certain common elements, or prototype patterns, that can be identified.
5. There is a small number of basic, primary, or prototype emotions.
6. All other emotions are mixed or derivative states; that is, they occur as combinations, mixtures, or compounds of the primary emotions.
7. Primary emotions are hypothetical constructs or idealized states whose properties and characteristics can only be inferred from various kinds of evidence.
8. Primary emotions can be conceptualized in terms of pairs of polar opposites.
9. All emotions vary in their degree of similarity to one another.
10. Each emotion can exist in varying degrees of intensity or levels of arousal.

Plutchik has represented the basic emotions in a wheel, represented in Figure 2.2, which, for each basic emotion, shows the opposite of it (e.g., joy vs. sadness), and mild and intense emotions related to it (e.g., for anger, mild would be annoyance and intense would be rage).

Ekman proposed another famous set of basic emotions, composed of six elements: anger, disgust, fear, happiness, sadness, and surprise [27]. This group of emotions was created

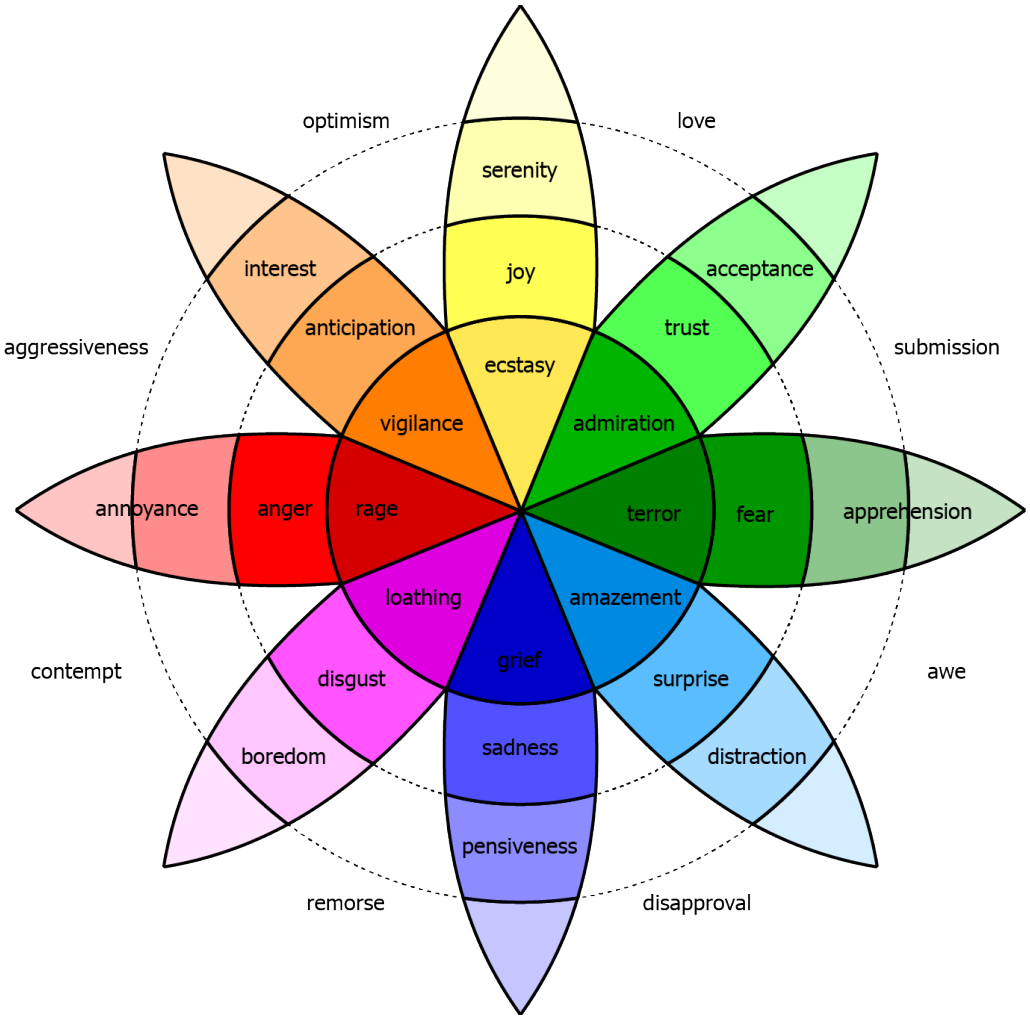


Figure 2.2: Plutchik's wheel of emotions, from [3].

to assess whether the relationship between facial expressions and emotions is distinct in different cultures; the research pointed that the relationship exists and is universal.

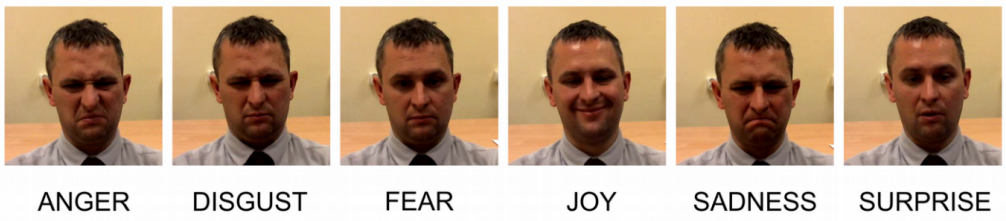


Figure 2.3: Ekman's six basic emotions associated with facial expressions, from [4].

Frijda proposed that emotions have two functions: the motivational and the informational functions. The motivational function focuses on the effects of the emotions in one's motivations. For example, a feeling of fear caused by an intimidating situation can motivate an individual to suppress thinking about it [9].

The informational function consists of giving adaptively useful information to other subsystems of personality. For instance, a pleasant feeling experienced when imagining a

possible course of action may signal the subconscious approval of the action. Also, the informational function of emotions can lead to "emotion-congruent" interpretations of ambiguous situations. For example, a subject can interpret ambiguous negative experiences in an angry way if feeling angry (e.g., to blame them on others) [9].

2.3 Knowledge Discovery in Databases

This project covers a problem of knowledge discovery, in particular, of pattern discovery to identify the personality of an individual through texts written by them. Knowledge discovery is, according to [5], "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data". The objective of this topic is to identify the personality of an individual (which is the potential result) through information gained (patterns) by way of processing the data retrieved.

Various authors define different groups of steps to describe the KDD process. In [5] are described nine basic steps that comprise the procedure: understand the application domain and identify the goal, choose the data set, data cleaning and preprocessing, data reduction and projection, matching the goal with an explicit data mining technique, creation of a hypothesis, data mining, interpretation of the results, and finally use of the new information. These steps are illustrated by Figure 2.4. The authors of [6] propose a simplified knowledge discovery procedure, which can be seen in Figure 2.5.

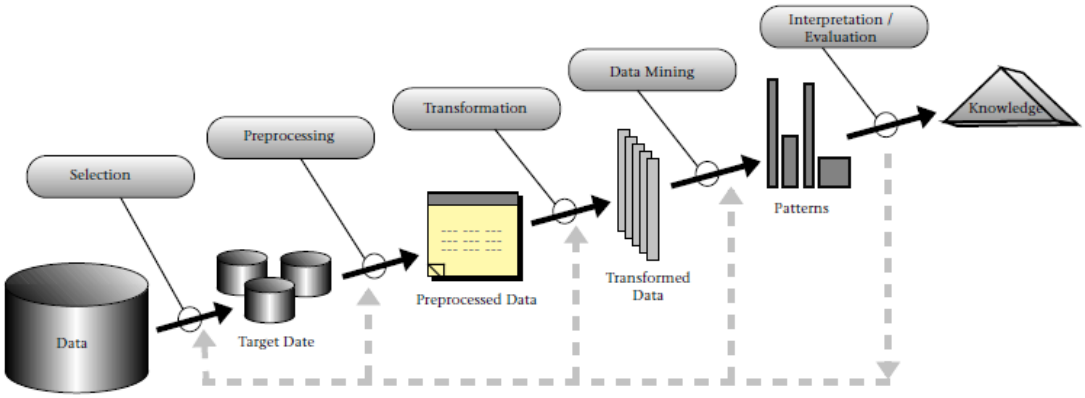


Figure 2.4: "An Overview of the Steps That Compose the KDD Process" [5]

The whole KDD operation is "interactive and iterative" [5], meaning that the outputs of a step can be used as the inputs of other steps, and you can go back to any previous step to further improve the discovery.

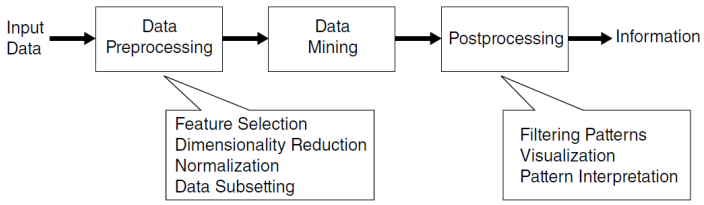


Figure 2.5: "The process of knowledge discovery in databases (KDD)" [6]

2.3.1 Data Preprocessing

The data preprocessing step involves several complex techniques with the goal of improving the data mining process in relation to time spent, cost, and quality of the analysis [6]. The subsequent sections will elaborate more on these methods.

2.3.1.1 Aggregation

Aggregation is "the combining of two or more objects into a single object" [6]. Consider a data set consisting of records of the daily weather: forecast, temperature, humidity, and cloud cover. One way to aggregate the data is to replace the individual records for a representative of each week.

Table 2.3: Example of a daily weather data set.

Date	Forecast	Temperature	Humidity	Cloud cover
2020/04/19	Sunny	25 °C	10 %	20 %
2020/04/20	Overcast	21 °C	50 %	80 %
2020/04/21	Raining	18 °C	90 %	90 %

Qualitative attributes are handled differently from quantitative attributes. Quantitative features "are typically aggregated by taking a sum or an average", while qualitative variables are usually "omitted or summarized" [6]. In the example shown in Table 2.3, the "temperature" variable could be represented by an average of the temperatures for the week, while the "forecast" attribute could be summarized into one of the labels, e.g., the one that most represents the weather for that week.

There are many advantages to aggregation, according to [6]:

1. "The smaller data sets resulting from data reduction require less memory and processing time, and hence, aggregation may permit the use of more expensive data mining algorithms";
2. Aggregating can provide a high-level perspective of the data by joining the multitude of low-level information;

3. It stabilizes the behavior of the data since aggregate values have less variability than their individual counterparts.

The main disadvantage of aggregation is the potential loss of valuable details, e.g., one of the lost details in the previous example is which day of the week the temperature peaked.

2.3.1.2 Sampling

Sampling is the process of choosing a subset to represent a larger portion of data in order to reduce the time and cost of the data processing.

The sample should be representative of the original data set, i.e. it should have roughly the same property of interest of the initial data set. For instance, "if the mean (average) of the data objects is the property of interest, then a sample is representative if it has a mean that is close to that of the original data" [6].

The easiest type of sampling is simple random sampling, in which individual items are equally probable of being selected from the dataset.

2.3.1.3 Dimensionality Reduction

Data sets can have a large number of features and, as such, suffer from the "curse of dimensionality" [6], which is the event of increasing the difficulty of the data analysis as the dimensionality of the data enlarges.

Reducing the data set dimensionality has several advantages. First, it can remove "irrelevant features and reduce noise" [6]. Second, it provides a more understandable model, which in turn allows a clearer visualization of the data. Finally, it also reduces the resources required by the data mining algorithm.

2.3.1.4 Feature Subset Selection

Selecting interesting attributes is a way of reducing the dimensionality of a data set. Features can be considered redundant if the information they provided is also given by one or more other features, or irrelevant if they do not supply useful information to the data mining process.

According to [6], "some irrelevant and redundant attributes can be eliminated immediately by using common sense or domain knowledge", however, frequently, it is necessary

to resort to a systematic approach to select the best subset of features.

There are three standard approaches to feature selection [6]:

- Embedded approaches: the data mining algorithm decides which subset of features to use - this is often the case with decision tree classifiers, described in Section 2.5.2;
- Filter approaches: the selection of the features is done using some technique that is independent of the data mining algorithm;
- Wrapper approaches: the data mining algorithm is fed with a large number of possible subsets in order to find the best subset of attributes, i.e. the subset for which the algorithm performs better.

An alternative to dimensionality reduction is feature weighting: features deemeed more important "are assigned a higher weight, while less important features are given a lower weight" [6].

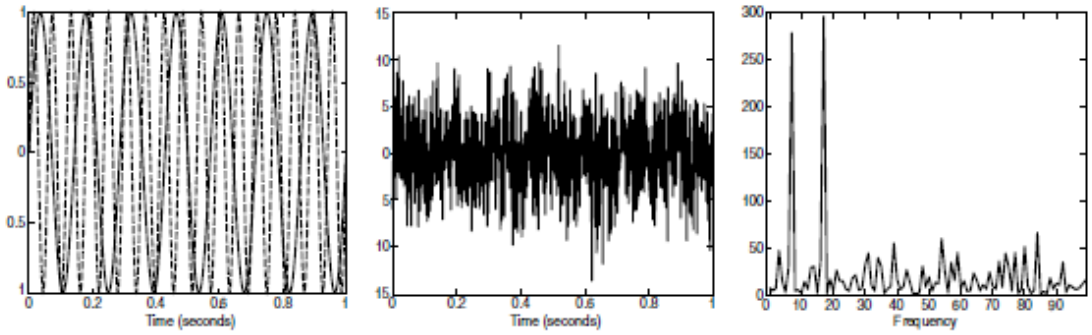
2.3.1.5 Feature Creation

Usually, new features can be created from the original set of attributes to condense the important information. According to [6], there are three procedures for creating new features: feature extraction, mapping the data to a new space, and feature construction.

"The creation of a new set of features from the original raw data is known as feature extraction" [6]. It is highly domain-specific, i.e. the techniques for extracting features are specific to a field and have limited applicability to other areas of knowledge, for instance image processing procedures may not be appropriate to natural language processing.

The second procedure, mapping the data to a new space, can be useful to reveal interesting features. Figure 2.6 illustrates the application of a Fourier transform to the noisy time series (the noisy time series being the sum of the first two time series with random noise), and shows that, despite of the noise, "there are two peaks that correspond to the periods of the two original, non-noisy time series" [6].

The third procedure is feature construction, and it consists in putting the already existing necessary information in a more suitable form for analysis. As an example, "consider a data set consisting of information about historical artifacts, which, along with other information, contains the volume and mass of each artifact" [6], and that we want to classify them by building material. A density feature would help produce an accurate classification, and it can be created from already existing features: $density = mass/volume$.



(a) Two time series. (b) Noisy time series. (c) Power spectrum

Figure 2.6: "Application of the Fourier transform to identify the underlying frequencies in time series data" [6].

2.3.1.6 Discretization and Binarization

Certain algorithms require the data to be in the form of categorical, or discrete, attributes, this is, variables that can take on one of a limited number of values, or binary attributes, this is, variables that can take on one of two values. It is often necessary to transform a continuous attribute into a discrete attribute using discretization, and both continuous and discrete features into binary features using binarization [6].

The simplest binarization technique of a group of n discrete values is to assign an integer in the interval $[0, n-1]$ to each of those values, and then convert the result to binary numbers [6]. Table 2.4 shows an example of a conversion of a categorical feature with five values to three binary attributes.

Table 2.4: "Conversion of a categorical attribute to three binary attributes" [6].

Categorical Value	**Integer Value**	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

This transformation, however, can cause some complications, "such as creating unintended relationships among the transformed attributes" [6]. For the example below, the attributes x_2 and x_3 are correlated because of the value "good". To avoid this issue, one binary attribute should be created for each categorical value, as shown in Table 2.5.

The discretization task consists of two steps. First, the values for the continuous feature are sorted and divided into n intervals by choosing $n-1$ split points. Second, all the values in one interval are mapped to the same discrete value [6]. The first step is where the problem of discretization is found: "deciding how many split points to choose and

Table 2.5: "Conversion of a categorical attribute to five asymmetric binary attributes" [6].

Categorical Value	**Integer Value**	x_1	x_2	x_3	x_4	x_5
awful	0	1	0	0	0	0
poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
good	3	0	0	0	1	0
great	4	0	0	0	0	1

where to place them".

The discretization process can be either supervised or unsupervised. Unsupervised discretization does not use the class information for inferring a discrete value, and is based on simple approaches, such as the equal width approach, in which the range of attribute values is divided in n intervals of equal width, or the equal frequency approach, which tries to create intervals with the same number of entries. Clustering methods can also be used for unsupervised discretization.

Supervised discretization, unlike unsupervised discretization, uses class information and knowledge about the field to obtain results. The main approaches are entropy-based, and a simple one would be to bissect "the initial values so that the resulting two intervals give minimum entropy", and then repeat the process with the interval with the highest entropy until reaching a stopping criterion [6].

2.3.1.7 Variable Transformation

According to [6], "a variable transformation refers to a transformation that is applied to all the values of a variable". Common types of variable transformation include the application of simple functions (e.g., $\log x$), and feature scaling functions such as normalization and standardization.

Feature scaling is the process of adjusting values on different scales to be represented in a common scale with the goal of avoiding having a variable with large values dominate the results of the calculation [6].

Normalization, also referred as min-max normalization or min-max scaling, consists in rescaling the values so they range in $[0, 1]$ or $[-1, 1]$ [28]. The formula for the $[0, 1]$ range is the following:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

where x is the original value and x' is the normalized value. To normalize a set of values

to $[a, b]$, the following formula is used:

$$x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}$$

Standardization is the process of rescaling the values so each value has zero-mean, i.e. a mean of 0, and unit-variance, i.e. a standard deviation of 1, and it is given by the following equation [28][6]:

$$x' = \frac{x - \bar{x}}{\sigma}$$

where x is the original value, $\bar{x}$ is the mean of the attribute values, σ is the standard deviation of the feature values, and x' is the standardized value.

2.3.2 Data Mining

Data mining is, in line with [6], "the process of automatically discovering useful information in large data repositories".

Data mining tasks are usually divided into two major categories [5][6]:

- **Predictive tasks**. "The objective of these tasks is to predict the value of a particular attribute based on the values of other attributes" [6].
- **Descriptive tasks**. The objective is to find patterns that allow the user to understand relationships in data.

This step of the KDD process "involves fitting models to, or determining patterns from, observed data" [5], mainly using the next two approaches:

- **Statistical**: allows for nondeterministic effects in the model (i.e. the model can perform differently on different runs);
- **Logical**: a purely deterministic approach.

The statistical approach tends to be more used since it takes into account the "uncertainty in real-world data-generating processes" [5] and, for that reason, most data mining methods are based on reliable techniques from machine learning, pattern recognition, and statistics.

2.3.3 Common Data Mining Methods

Classification is described in [6] as "the task of assigning objects to one of several predefined categories". Our work focuses on this method, and in Section 2.5 are detailed machine learning algorithms for classification purposes. Figure 2.7 shows the result of linear classification into two categories.

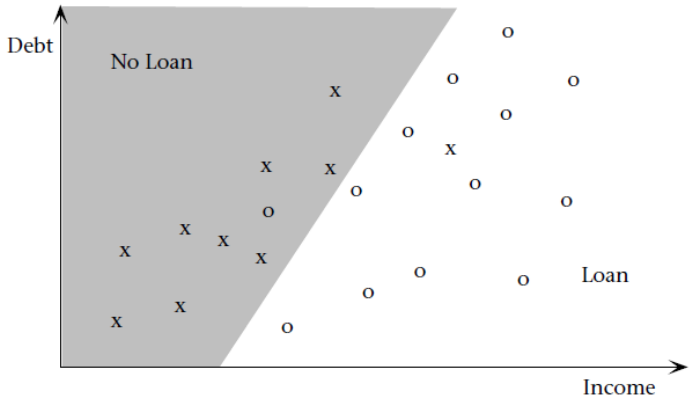


Figure 2.7: "A Simple Linear Classification Boundary for the Loan Data Set" [5].

Regression is the task of finding a correlation between a dependent (or response, outcome) variable and one or more independent (or predictor, feature) variables. Linear regression is the most common type of regression analysis, and its goal is to find a linear combination that most closely fits the data. Figure 2.8 "shows the result of simple linear regression where total debt is fitted as a linear function of income" [5].

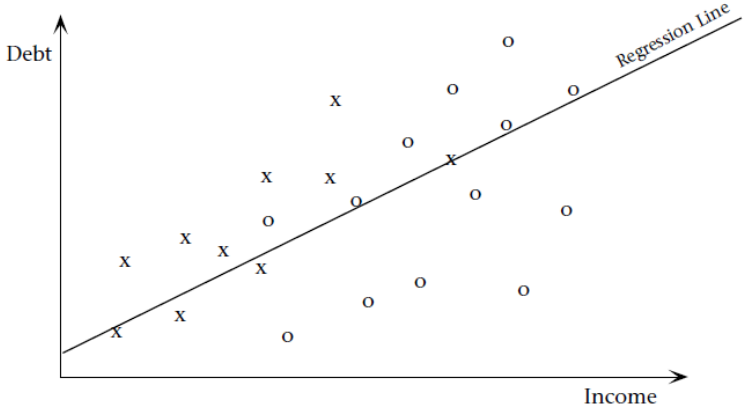


Figure 2.8: "Simple Linear Regression for the Loan Data Set" [5].

Clustering seeks to divide data into groups (clusters) in such way that objects inside one cluster are more similar (given some criteria) to each other than to those in other clusters [6][29].

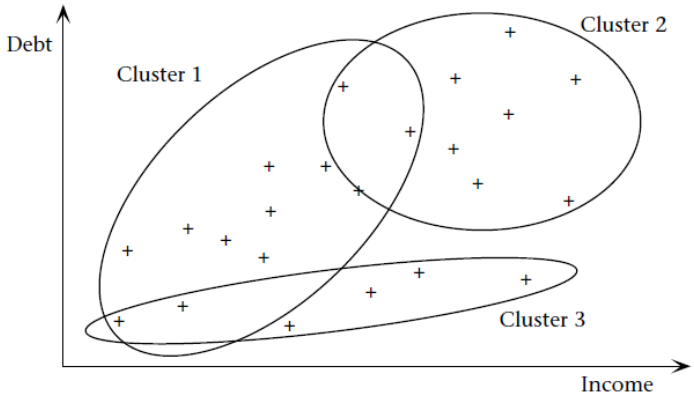


Figure 2.9: “A Simple Clustering of the Loan Data Set into Three Clusters” [5].

2.3.4 Application Issues

The authors of [5] present a list of challenges in KDD application, ranging from issues caused by the properties of data sets (such as size or noisiness) to real-world changes (for instance, new advancements in the field of knowledge). The following list presents some of those points.

- **High dimensionality**. A high-dimensional data set requires more resources for the data mining algorithm, and may lead the algorithm to find patterns that are not valid in general. This issue can be tackled using dimensionality reduction techniques, which are described in Section 2.3.1.3 of this work;
- **Overfitting**. Overfitting is ”the production of an analysis which corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably” [30]. One usual solution to overfitting is cross-validation - this topic is discussed in more detail in Section 2.3.5.
- **Changing data and knowledge**. Nonstationary data can make previously discovered patterns invalid. Incremental methods for updating the patterns can be a solution to this problem.
- **Missing and noisy data**. Procedures to identify missing variables and remove noise should be used. Feature subset selection, described in 2.3.1.4, is one technique to irrelevant and redundant variables.
- **Complex relationships between fields**. Not all algorithms are capable of analyzing data sets with relations between attributes or values, thus techniques for deriving relations between variables should be used.
- **Understandability of patterns**. It may be hard for humans to understand the patterns machine learning algorithms discover. Techniques for data visualization,

such as graphic representations or natural language generation, allow for better understandability. One practical example is the graph generation for decision tree algorithms.

- **User interaction and prior knowledge**. Some methods require previous knowledge to work properly - for instance, Bayesian processes use preceding probabilities for their goal.

2.3.5 Postprocessing

Postprocessing consists of tasks and methods to further process the knowledge derived from the data mining step. Usually, these procedures are categorized into knowledge filtering, interpretation and explanation, evaluation, and knowledge integration [31].

Knowledge filtering is useful when the data mining algorithm induces rules that only apply to an insignificant number of training samples. Assume there is a decision tree that was generated with a noisy training data set, creating leaves that cover a small number of cases. In this situation, pruning the tree could prove helpful.

Interpretation and explanation focuses on obtaining a human-understandable view of the data mining results, e.g., generating a decision tree graph representation, and on interpreting the results matching them with domain-specific knowledge.

Knowledge integration is the combination of results from several models to further increase the accuracy and success rates of the data mining process.

Evaluation takes place after the data mining phase to assess the validity of the results. As stated in [6], the "evaluation of the performance of a classification model is based on the counts of test records correctly and incorrectly predicted by the model", and these counts are usually represented in the form of a confusion matrix. Table 2.6 shows a confusion matrix for a 2-class problem, where each entry f_{ij} is the number of records from class i predicted to be of class j.

Table 2.6: "Confusion matrix for a 2-class problem" [6].

		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

The values a confusion matrix provides can be used to calculate performance metrics, such as accuracy and error rate.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

$$Error\ rate = \frac{Number\ of\ wrong\ predictions}{Total\ number\ of\ predictions}$$

For the example in Table 2.6, the equations for the accuracy and error rate can be written as the following:

$$Accuracy = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

$$Error\ rate = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

Precision, recall, and F1 score are other metrics that are used often. Precision is the ratio of correct positive predictions to the total of positive predictions, meaning that a high precision value relates to a low number of false positives. Recall or sensitivity is the ratio of correct predictions of a class to the actual number of records of that class, being useful to observe how many records of a determined class the model correctly predicts. F1 score is another measure of a model's accuracy and ranges between 0 and 1, the latter meaning perfect precision and recall.

$$Precision = \frac{Number\ of\ correct\ positive\ predictions}{Total\ number\ of\ positive\ predictions}$$

$$Recall = \frac{Number\ of\ correct\ positive\ predictions}{Number\ of\ actual\ positive\ records}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Most classification algorithms seek models with the highest accuracy metric [6]. However, there may be cases where the model fits the data set too well, adjusting even to the noise of it, incurring in a problem of overfitting (described in 2.3.4).

Validation of the model should be performed for determining if the patterns found are valid in general. The most common approach to validate a model is cross-validation. Assume there is a data set and that we split it into two equal-sized subsets. First, we use one of the partitions for training and the other for testing the model. Then, the partitions

are switched so that the training set is now the testing set and vice versa [6]. This approach is called two-fold cross-validation. Cross-validation is also referred to as k-fold cross-validation because it can be performed with any k number of partitions.

2.4 Natural Language Tools and Lexicons

This section starts by introducing fundamental concepts of linguistics in Section 2.4.1. Then, the concept of word embedding is briefly explained in Section 2.4.2. Sections 2.4.3 to 2.4.8 introduce natural language databases and lexicons, namely LIWC, WordNet, NRC Emotion Lexicon, MRC Psycholinguistic Database, EmoSenticNet, and ConceptNet. Ultimately, in Section 2.4.9 we present the NLTK library for Python and show some examples of usage.

2.4.1 Introduction to Linguistics

Linguistics, the scientific study of language, consists of several subfields, such as phonetics, phonology, morphology, lexicon, syntax, semantics, and pragmatics [32]. Phonetics and phonology deal with how words sound, and as such, they are out of the scope of this work.

Syntax sets the rules and principles that specify the structure of sentences [33]. For instance, both English and Portuguese languages follow a subject-verb-object order (e.g., "I will buy a house", "eu vou comprar uma casa"), whereas the Japanese syntax states that sentences follow the subject-order-verb sequence.

Morphology researches the structure of words and their integrant parts [33]. Through morphology, we can understand how words are formed and how they relate to each other. The elemental parts of a word consist of root words, stems, prefixes and suffixes. A root word is the indivisible core of a word, which means it is its primary lexical unit (e.g., "appear"). Stems are forms of a word to which we can attach affixes, and can be composed of one or more root words (e.g., "reappear" is the stem of "reappearance" and "reappeared", but is not a root word). Prefixes and suffixes are affixes that are added before or after a stem, respectively (e.g., "*un*happy", "work*ed*").

Lexicon is the set of information about properties of words, such as what part-of-speech a word is (e.g., the word "well" is a noun in "throw a coin into the well", an interjection in "well, it was canceled", an adverb in "she speaks well", and an adjective in "he is well'), irregular morphological forms (e.g., "feet", not "foots"), how a word is pronounced (e.g., "paper" is pronounced as /ˈpeɪ.pər/ in American English), among others [32].

Semantics and pragmatics both deal with the meaning of words and sentences. While semantics is "about the manner in which lexical meaning is combined morphologically and syntactically to form the meaning of a sentence", pragmatics is about the situations "where context includes both the linguistic and situational context" of a sentence, e.g., if someone says "lovely day" on a rainy day, we use the general knowledge that people usually do not like rain to infer that they are being ironic [32].

2.4.2 Word Embedding

Word embedding is a language modelling technique in which words are mapped to vectors of real numbers. Simple language modelling methods treat words as atomic units and, consequently, there is no notion of similarity between words [11]. Word embedding tackles this problem by providing distributional characteristics through the mapping of lexical items on a semantic space [34]. One of the most known word embedding algorithms is word2vec, proposed in [11]. Table 2.7 shows some of the relationships between the words of word2vec's data set.

Table 2.7: Examples of relationships between words on word2vec's data set, from [11].

Type of relationship	**Word Pair 1**		**Word Pair 2**	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Since the words are transformed into vectors of numbers, it is possible to apply arithmetic operations to them. The most common way to calculate the similarity between the vectors, i.e. how closely related words are, is by calculating the cosine similarity. Let two vectors u and v represent respectively the words w_u and w_v, the cosine similarity between the two words can be computed as follows:

$$similarity(w_u, w_v) = cos(\theta) = \frac{u \cdot v}{||u||\,||v||} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}$$

Table 2.8 presents some examples of calculated similarities between words of the word2vec's

data set.

Table 2.8: Examples of similarity between words from word2vec's data set.

Word 1	Word 2	Similarity
car	sky	0.087
house	dog	0.257
dog	cat	0.761

2.4.3 Linguistic Inquiry and Word Count

In 1996, James W. Pennebaker and Martha E. Francis tried to prove and explain how "writing about upsetting experiences can improve physical health" [35] and, for that effect, created and validated a text analysis tool to which they gave the name Linguistic Inquiry and Word Count, or LIWC.

LIWC is composed of two complementary parts: a text processing module and a support dictionary [35]. The text processing module reads and analyzes a given content using the backing dictionary. The dictionary is a collection of words categorized into dimensions of language, such as optimism or tentativeness. The latter component was used here in our work.

Both LIWC components were updated over the years and, as of this writing, the latest version is the LIWC2015. The LIWC2015 English dictionary is made of 6548 words and stems, annotated within 73 categories and subcategories, which are listed in Table 2.9. The authors state that the latest dictionary "captures, on average, over 86 percent of the words people use in writing and speech" [1].

The LIWC2015 dictionary has been created through a series of steps that start with the word collection. Starting with the LIWC2007 dictionary, a group of judges generated word lists from a variety of sources and then brain-stormed to see which ones were relevant. Next, the authors created a working dictionary from these words and checked the frequency dictionary words were used in diverse contexts. After this, they identified words that appeared frequently in corpora that were not already on the list. Following these steps, a psychometric evaluation was done to assess the reliability and validity of the generated dictionary. Finally, the whole process was repeated to catch any possible mistakes [1].

LIWC's text processing module's output is composed of the word count of the input text, the number of words per sentence, the percentage of words of each category, the percentage of words with more than six letters, the percentage of words found in the dictionary, and four summary variables (analytical thinking, clout, authenticity, and emotional tone). These summary variables are non-transparent, and "derived from previously published findings from our lab and converted to percentiles based on standardized scores from

large comparison samples" [1].

Table 2.9: List of categories on LIWC2015, adapted from [1].

Category	Abbrev	Examples
Linguistic Dimensions		
Total function words	funct	it, to, no, very
Total pronouns	pronoun	I, them, itself
Personal pronouns	ppron	I, them, her
1st pers singular	i	I, me, mine
1st pers plural	we	we, us, our
2nd person	you	you, your, thou
3rd pers singular	shehe	she, her, him
3rd pers plural	they	they, their, they'd
Impersonal pronouns	ipron	it, it's, those
Articles	article	a, an, the
Prepositions	prep	to, with, above
Auxiliary verbs	auxverb	am, will, have
Common adverbs	adverb	very, really
Conjunctions	conj	and, but, whereas
Negations	negate	no, not, never
Other Grammar		
Common verbs	verb	eat, come, carry
Common adjectives	adj	free, happy, long
Comparisons	compare	greater, best, after
Interrogatives	interrog	how, when, what
Numbers	number	second, thousand
Quantifiers	quant	few, many, much
Psychological Processes		
Affective processes	affect	happy, cried
Positive emotion	posemo	love, nice, sweet
Negative emotion	negemo	hurt, ugly, nasty
Anxiety	anx	worried, fearful
Anger	anger	hate, kill, annoyed
Sadness	sad	crying, grief, sad
Social processes	social	mate, talk, they
Family	family	daughter, dad, aunt
Friends	friend	buddy, neighbor
Female references	female	girl, her, mom
Male references	male	boy, his, dad
Cognitive processes	cogproc	cause, know, ought
Insight	insight	think, know

Category	Abbrev	Examples
Causation	cause	because, effect
Discrepancy	discrep	should, would
Tentative	tentat	maybe, perhaps
Certainty	certain	always, never
Differentiation	differ	hasn't, but, else
Perceptual processes	percept	look, heard, feeling
See	see	view, saw, seen
Hear	hear	listen, hearing
Feel	feel	feels, touch
Biological processes	bio	eat, blood, pain
Body	body	cheek, hands, spit
Health	health	clinic, flu, pill
Sexual	sexual	horny, love, incest
Ingestion	ingest	dish, eat, pizza
Drives	drives	
Affiliation	affiliation	ally, friend, social
Achievement	achieve	win, success, better
Power	power	superior, bully
Reward	reward	take, prize, benefit
Risk	risk	danger, doubt
Time orientations	timeorient	
Past focus	focuspast	ago, did, talked
Present focus	focuspresent	today, is, now
Future focus	focusfuture	may, will, soon
Relativity	relativ	area, bend, exit
Motion	motion	arrive, car, go
Space	space	down, in, thin
Time	time	end, until, season
Personal concerns		
Work	work	job, majors, xerox
Leisure	leisure	cook, chat, movie
Home	home	kitchen, landlord
Money	money	audit, cash, owe
Religion	relig	altar, church
Death	death	bury, coffin, kill
Informal language	informal	
Swear words	swear	fuck, damn, shit
Netspeak	netspeak	btw, lol, thx
Assent	assent	agree, OK, yes
Nonfluencies	nonflu	er, hm, umm

Category	Abbrev	Examples
Fillers	filler	Imean, youknow

2.4.4 WordNet

WordNet is a lexical database that links words into sets of cognitive synonyms, or synsets, each expressing a single concept [36]. Synsets are connected to each others through conceptual-semantic and lexical relations.

Noun

S: (n) **dog**, domestic dog, Canis familiaris (a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds) *"the dog barked all night"*
S: (n) frump, **dog** (a dull unattractive unpleasant girl or woman) *"she got a reputation as a frump"; "she's a real dog"*
S: (n) **dog** (informal term for a man) *"you lucky dog"*
S: (n) cad, bounder, blackguard, **dog**, hound, heel (someone who is morally reprehensible) *"you dirty dog"*
S: (n) frank, frankfurter, hotdog, hot dog, **dog**, wiener, wienerwurst, weenie (a smooth-textured sausage of minced beef or pork usually smoked; often served on a bread roll)
S: (n) pawl, detent, click, **dog** (a hinged catch that fits into a notch of a ratchet to move a wheel forward or prevent it from moving backward)
S: (n) andiron, firedog, **dog**, dog-iron (metal supports for logs in a fireplace) *"the andirons were too hot to touch"*

Verb

S: (v) chase, chase after, trail, tail, tag, give chase, **dog**, go after, track (go after with the intent to catch) *"The policeman chased the mugger down the alley"; "the dog chased the rabbit"*

Figure 2.10: WordNet's synsets for the word "dog".

The WordNet lexicon resembles both a thesaurus and a dictionary [37]. It can be used to find words that express a specific concept by searching up a word that articulates that concept, thus working as a thesaurus. One advantage that WordNet has over thesaurus is that the relationships between words are explicit and labeled, which allows for the user to select the relation that is most adequate to their conceptual space. WordNet can also be perceived as a dictionary as it "gives definitions and sample sentences for most of its synsets" [37]. Figure 2.10 shows WordNet's results for the word "dog". Note that that each synset is annotated with related words, its definition, and example phrases.

2.4.5 NRC Emotion Lexicon

The NRC Emotion Lexicon is a list of more than 14000 words and their associations with the eight emotions from Plutchik's model, referred in Section 2.2.3, and two sentiments (positive and negative) [38].

Table 2.10: Sample annotations from NRC Emotion Lexicon.

Word	Positive	Negative	Anger	Anticipation	Disgust	Fear	Joy	Sadness	Surprise	Trust
hilarious	1	0	0	0	0	0	1	0	1	0
petroleum	1	1	0	0	1	0	0	0	0	0
scapegoat	0	1	1	0	0	1	0	0	0	0

This lexicon was crowdsourced on Amazon's Mechanical Turk by having humans annotating words with the emotions and sentiments that they would find adequate.

The investigators found some issues with crowdsourcing and emotion annotation [38]. On the crowdsourcing platform side, the task can attract cheaters (who may input random information) or malicious annotators (who may input wrong information deliberately).

On the emotion annotation part, there are linguistic barriers that can lead to issues. One potential issue they found is that native and fluent speakers are better at identifying emotions in text, so they require the annotators to be native or fluent in English. The other problem discussed is that words can evoke different emotions if used in different contexts.

To tackle these topics, the author presented the annotators with a word choice challenge. They would have to choose the word that would better fit into the target, out of four options. From the four words, three would be "irrelevant distractors" [38]. If this challenge was not answered correctly, the annotation would be discarded.

2.4.6 MRC Psycholinguistic Database

The MRC Psycholinguistic Database is a machine usable dictionary file, according to [39]. This file contains over 150000 words and provides information about 26 different linguistic attributes. Table 2.11 lists the properties in the dictionary.

The author states that it differs from other datasets in that "it includes not only syntactic information but also psychological data for the entries", and in that it does not provide any semantic information [39].

As of this writing, the database is available as an online service on `https://websites.psychology.uwa.edu.au/school/MRCDatabase/uwa_mrc.htm`.

2.4.7 EmoSenticNet

EmoSenticNet is based on two datasets [40]:

- SenticNet - it "provides a set of semantics, sentics, and polarity associated with

Table 2.11: Composition of MRC's dictionary file.

Name	Property
NLET	Number of letters in the word
NPHON	Number of phonemes in the word
NSYL	Number of syllables in the word
K-F-FREQ	Kucera and Francis written frequency
K-F-NCATS	Kucera and Francis number of categories
K-F-NSAMP	Kucera and Francis number of samples
T-L-FREQ	Thorndike-Lorge frequency
BROWN-FREQ	Brown verbal frequency
FAM	Familiarity
CONC	Concreteness
IMAG	Imagery
MEANC	Mean Colerado Meaningfulness
MEANP	Mean Pavio Meaningfulnesss
AOA	Age of Acquisition
TQ2	Type
WTYPE	Part of Speech
PDWTYPE PD	Part of Speech
ALPHSYL	Alphasyllable
STATUS	Status
VAR	Varient Phoneme
CAP	Written Capitalised
IRREG	Irregular Plural
WORD	The actual word
PHON	Phonetic Transcription
DPHON	Edited Phonetic Transcription
STRESS	Stress Pattern

200,000 natural language concepts" [41] - the polarity scores range between -1 (extremely negative) and 1 (extremely positive);

- WordNet-Affect (WNA) Emotion Lists - it assigns Ekman's six basic emotions (fear, anger, joy, sadness, disgust, and surprise, explained in more detail in 2.2.3) to synsets of WordNet (see 2.4.4).

The authors of EmoSenticNet started from the premise of creating a methodology to "automatically assign emotion labels to SenticNet concepts" by training a classifier on SenticNet concepts that are present in the WNA database. Table 2.12 shows some examples of concepts with the six emotion annotations.

Table 2.12: Samples from the EmoSenticNet lexicon.

Concepts	Anger	Disgust	Joy	Sad	Surprise	Fear
park_ticket	1	0	0	0	0	0
catch_bus	0	0	1	0	0	0
remove_stain	0	1	0	0	0	0

2.4.8 ConceptNet

ConceptNet is "a knowledge graph that connects words and phrases of natural language (terms) with labeled, weighted edges (assertions)", according to [42]. It represents relations between words, such as "*dog* is capable of *guard your house*" or "*bicycle* is a type of *transportation*".

There are 36 core relations between terms [42]:

- **Symmetric relations**: *Antonym*, *DistinctFrom*, *EtymologicallyRelatedTo*, *LocatedNear*, *RelatedTo*, *SimilarTo*, and *Synonym*
- **Asymmetric relations**: *AtLocation*, *CapableOf*, *Causes*, *CausesDesire*, *CreatedBy*, *DefinedAs*, *DerivedFrom*, *Desires*, *Entails*, *ExternalURL*, *FormOf*, *HasA*, *HasContext*, *HasFirstSubevent*, *HasLastSubevent*, *HasPrerequisite*, *HasProperty*, *InstanceOf*, *IsA*, *MadeOf*, *MannerOf*, *MotivatedByGoal*, *ObstructedBy*, *PartOf*, *ReceivesAction*, *SenseOf*, *SymbolOf*, and *UsedFor*

This network contains over 21 million edges and more than 8 million nodes, with information in, at least, 83 languages. The largest sources of knowledge are Wiktionary and Open Mind Common Sense.

2.4.9 Natural Language Toolkit

The Natural Language Toolkit is a free, open source suite of Python modules and data sets to work with human language. According to [43], NLTK provides interfaces to "over 50 corpora and lexical resources", and libraries for "classification, tokenization, stemming, tagging, parsing, and semantic reasoning".

NLTK supplies an interface to download corpora, such as the WordNet corpus, as seen in Figure 2.11.

This library implements several functions for tokenization. For instance, we can use the default `word_tokenize` if we do not need any configuration, or have total control over the tokenization with an instance of `RegexpTokenizer` which accepts a regular expression. Listing 1 presents the difference between these two tokenization methods.

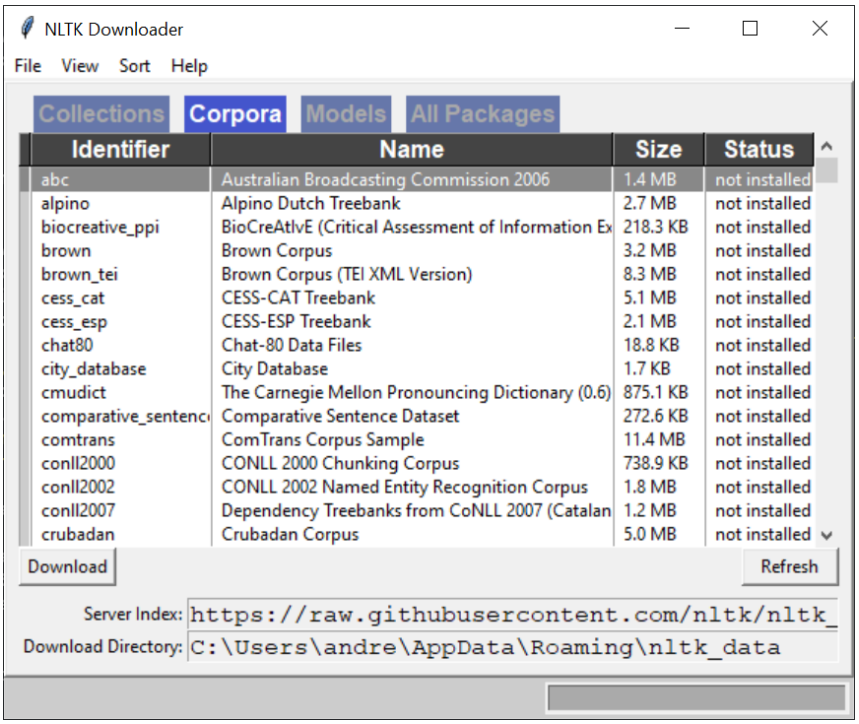


Figure 2.11: Screenshot of the NLTK Downloader.

Listing 1 Example of NLTK's tokenization capabilities.

```
>>> sentence = "I didn't expect this outcome."
>>> nltk.word_tokenize(sentence)
['I', 'did', "n't", 'expect', 'this', 'outcome']
>>> from nltk.tokenize import RegexpTokenizer
>>> RegexpTokenizer(r"\w+(?:['-]\w+)?").tokenize(sentence)
['I', "didn't", 'expect', 'this', 'outcome']
```

There are also functions for stemming words, such as the `PorterStemmer`, which is based on Porter's stemming algorithm described in [44]. Listing 2 shows examples of this stemmer's output.

Listing 2 Example of NLTK's stemming functionality.

```
>>> from nltk.stem import PorterStemmer
>>> PorterStemmer().stem("consulting")
'consult'
>>> PorterStemmer().stem("reappearance")
'reappear'
```

NLTK provides a WordNet interface which we can use to find WordNet synsets. It allows the specification of several parameters, such as part of speech and language (the latter requires downloading the Open Multilingual WordNet corpus). For specific synsets, we can also consult their definitions, hypernyms and hyponyms, and calculate their similarity. Examples of usage can be consulted in Listing 3. The official documentation on the WordNet interface can be consulted in [45].

Listing 3 Example of using the WordNet features through the NLTK interface.

```
>>> from nltk.corpus import wordnet
>>> # Find all synsets for the word "sugar"
>>> wordnet.synsets("sugar")
[Synset('sugar.n.01'), Synset('carbohydrate.n.01'), Synset('boodle.n.01'),
 Synset('sugar.v.01')]
>>> # Find synsets for the word "sugar" as a noun
>>> wordnet.synsets("sugar", wordnet.NOUN)
[Synset('sugar.n.01'), Synset('carbohydrate.n.01'), Synset('boodle.n.01')]
>>> # Find synsets for a word in a language other than english
>>> wordnet.synsets("açúcar", lang="por")
[Synset('sugar.n.01'), Synset('carbohydrate.n.01')]
>>> # Consult the definition for a specific synset
>>> wordnet.synset("sugar.n.01").definition()
'a white crystalline carbohydrate used as a sweetener and preservative'
>>> # Find hypernyms and hyponyms
>>> wordnet.synset("dog.n.01").hypernyms()
[Synset('canine.n.02'), Synset('domestic_animal.n.01')]
>>> wordnet.synset("dog.n.01").hyponyms()
[Synset('basenji.n.01'), Synset('corgi.n.01'), Synset('cur.n.01'),
 Synset('dalmatian.n.02'), Synset('great_pyrenees.n.01'), ...]
>>> # Calculate the similarity of two synsets
>>> wordnet.synset("dog.n.01").path_similarity(wordnet.synset("cat.n.01"))
0.2
```

2.5 Machine Learning Algorithms and Tools

In this section, we introduce four distinct machine learning algorithms for classification, specifically Naive Bayes (2.5.1), Decision Tree (2.5.2), Support-Vector Machine (2.5.3, and Random Forest (2.5.4). At last, in Section 2.5.5 we detail Scikit-learn, a machine learning suite for Python, complete with examples.

2.5.1 Naive Bayes Classifier

The Naive Bayes classifier is an algorithm that infers classes based on probabilities. The Bayes' theorem is the foundation of the Bayesian learning processes [7], and its equation is:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where

- A and B are events;

- $P(B) \neq 0$;
- $P(A)$ and $P(B)$ are the probabilities of observing the events A and B, respectively;
- $P(A|B)$ is the conditional probability of observing the event A given that the event B has occurred.

This classifier is useful to solve problems where each instance x is described by a number of features and the target function $f(x)$ can return any value from a finite set V. The algorithm must be trained using a chosen set of training samples. After training, the classifier can be asked to predict the classification of a new case, attributing the most probable target result, v, for the feature values $(a_1, a_2, ..., a_n)$.

$$v = \underset{v_j \in V}{argmax}\ P(v_j \mid a_1, a_2, ..., a_n)$$

Using the Bayes' theorem, the previous equation can be rewritten as:

$$v = \underset{v_j \in V}{argmax}\ \frac{P(a_1, a2, ..., a_n \mid v_j)\ P(v_j)}{P(a_1, a2, ..., a_n)}$$

The Naive Bayes classifier is based on the assumption that feature values are independent from the target value and its equation can be simplified as such:

$$v = \underset{v_j \in V}{argmax}\ P(v_j) \prod_i P(a_i|v_j)$$

2.5.2 Decision Tree Learning

Decision tree learning is a predictive modelling approach based on the use of decision trees to infer discrete values [7]. The resulting target functions, or learned trees, can be represented as if-then rules to ease the human comprehension.

The classification is done by inquiring a sample from the root until it reaches a leaf of the tree. The Figure 2.12 shows a typical decision tree for whether the weather is suitable for playing tennis. For instance, the test case $Outlook = Rain,\ Wind = Strong$ would result in No.

This algorithm is useful for classification problems where there is a finite number of classes, and where the instances' attributes can assume values only from a finite set.

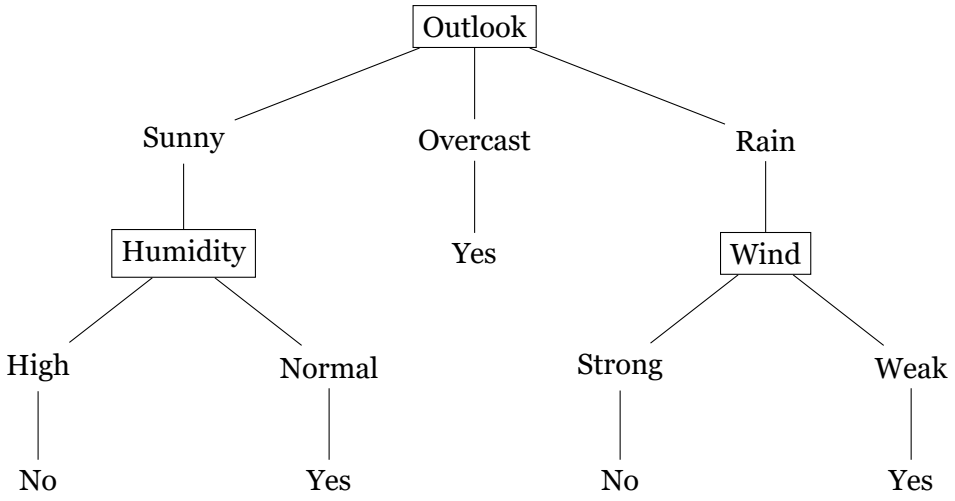


Figure 2.12: A decision tree for the concept *PlayTennis*, from [7].

To create the decision nodes, a metric called information gain must be calculated for each of the instances' attributes. Information gain is how much the entropy caused by splitting the training data set will be reduced. The calculation of the information gain can be described by the following equation:

$$Information\ Gain = Entropy(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where S is a group of examples, $V(A)$ is the set of values for the attribute A, and S_v is the subset of S for which the attribute A has value v.

The information entropy, or Shannon's entropy, concept coined by Claude Shannon in 1948 in his article "A Mathematical Theory of Communication" [46], can be determined by the following equation:

$$Entropy(S) = -\sum_{i=1}^{n} p_i \, log_b(p_i)$$

where S is an array of samples, n is the number of different values the class attribute can take, p_i is the proportion of S that belongs to class c, and b is the base of the logarithm used (usually 2, the Euler's number e, or 10).

2.5.3 Support-Vector Machine

Support-vector machines, or support-vector networks [8], are learning models in which the input are vectors with n dimensions, and those vectors are non-linearly mapped to a

high-dimension feature space. Then, a decision surface is built with high generalization ability in mind.

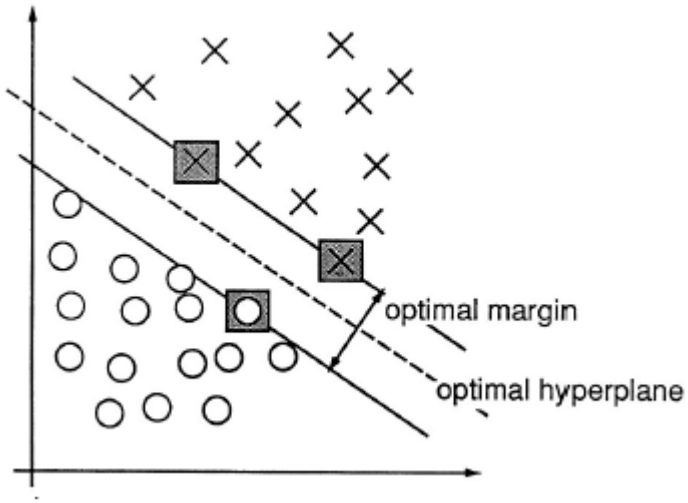


Figure 2.13: An example borrowed from [8] of a linear decision surface which separates instances in a 2 dimensional space to illustrate the underlying concept of a SVM.

For a linear problem, assume that there is a set of n training vectors

$$(x_1, y_1), ..., (y_n, x_n), \quad y_i \in -1, 1$$

which can be linearly separated if a vector w and a scalar b exist such that the inequalities

$$\begin{aligned} w \cdot x_i + b \geq 1 \quad if \quad y_i = 1, \\ w \cdot x_i + b \geq -1 \quad if \quad y_i = -1, \end{aligned}$$

are valid for all elements of the set. We can combine the two constraints into the following inequality:

$$y_i(w \cdot x_i + b) \geq 1, \quad i = 1, ..., n \, .$$

“The optimal hyperplane

$$w_0 \cdot x + b_0 = 0$$

is the unique one that separates the training vectors with a maximal margin: it determines

the direction $w/|w|$ where the distance between the projections of the training vectors of two different classes is maximal" [8]. The distance $\rho(w, b)$ is given by the following equation:

$$\rho(w, b) = \min_{\{x:y=1\}} \frac{x \cdot w}{|w|} - \max_{\{x:y=-1\}} \frac{x \cdot w}{|w|}$$

Optimal hyperplane's arguments (w_0, b_0) maximize this distance. It follows from the previous inequalities, distance equation, and dot product of two vectors that:

$$\rho(w_0, b_0) = \frac{2}{|w_0|} = \frac{2}{\sqrt{w_0 \cdot w_0}} .$$

The full demonstration of this formula can be consulted in Appendix A.1

2.5.4 Random Forest

A random forest, or random decision forest, is an ensemble classification algorithm, i.e. it uses multiple classification algorithms for a better performance. The random forest classificator generates a group of decision trees, which are described in Section 2.5.2, using the training data and outputs a statistic of the decision trees' predictions as its classification.

Random forests proposed to correct the common overfitting problem of the decision tree learning algorithm [47]. Overfitting a common data mining issue, referred as such in Section 2.3.4. In the case of decision trees, they become overfitted modules if the trees grow too complex.

The method proposed in [47] is to build decision trees from randomly selected subsets of the feature space using the whole training set. A discriminant function is then used to combine the classification of the decision trees. For a sample x, let $v_i(x)$ be the result of a tree $T_i(i = 1, ..., t)$. The probability that x belongs to class $c(c = 1, ..., n)$ is given by

$$P(c|v_i(x)) = \frac{P(c, v_i(x)}{\sum_{j=1}^{n} P(c_j, v_i(x))} ,$$

which is the fraction of all samples belonging to c over all instances assigned to $v_i(x)$. The discriminant function's equation is

$$f_c(x) = \frac{1}{t} \sum_{i=1}^{t} P(c|v_i(x))$$

and the decision rule is to assign the instance x to the class c for which $f_c(x)$ is maximum.

2.5.5 Scikit-learn

Scikit-learn, also known as sklearn, is a free, open-source machine learning module for the Python programming language. This library features algorithms for both supervised and unsupervised learning, model evaluation and validation, feature extraction, among others. We used scikit-learn in our work to extract features from the dictionaries, to classify the data set, and to validate and evaluate the derived models. The following paragraphs illustrate these scenarios.

Assume you have a dataset for the concept *PlayTennis*, the same in Figure 2.12, which consists in a list of objects with three properties ("outlook", "humidity" and "wind") and an outcome that can either be "yes" or "no". An example of this type of object is in Listing 4.

Listing 4 Example object of a dataset to illustrate the concept *PlayTennis*

```
{
  "outlook": "sunny",
  "humidity": "high",
  "wind": "weak"
}
```

We can then extract features from the dictionaries using a `DictVectorizer`, which transforms the objects to vectors (Listing 5). Setting the parameter `sparse` to `False` on the transformer's constructor makes the vectorizer return normal arrays instead of sparse matrices from the module `scipy`.

Then, we would want to split the data in two groups: the train set and the test set. For that purpose, we can use the function `train_test_split` as seen in Listing 6.

The function `train_test_split` accepts three parameters in this example: `x` is the list of samples to split, `y` is the class of each sample of `x`, `test_size` is the fraction of the data set that is assigned to the test set (in this case, 10%).

Next, we can classify the data set using a classifier. Scikit-learn provides many wrapper classes for different classifiers, such as `MultinomialNB` for Naive Bayes, `DecisionTreeClassifier` for decision trees, `SVC` for support-vector machines, and `RandomForestClassifier`

Listing 5 Example of feature extraction using a *DictVectorizer*.

```
>>> from sklearn.feature_extraction import DictVectorizer
>>> obj = {
...     'outlook': 'sunny',
...     'humidity': 'high',
...     'wind': 'weak'
... }
>>> obj2 = {
...     'outlook': 'rain',
...     'humidity': 'high',
...     'wind': 'strong'
... }
>>> x = [obj, obj2]
>>> dv = DictVectorizer(sparse=False)
>>> dv.fit_transform(x)
array([[1., 0., 1., 0., 1.],
       [1., 1., 0., 1., 0.]])
```

Listing 6 Example of data set splitting using `train_test_split`.

```
from sklearn.model_selection import train_test_split
x = [object1, object2]
y = ['yes', 'no']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1)
```

for random forests. These classifiers are highly parameterizable, and well documented in [48]. Listing 7 shows the test set classification using a decision tree classifier fitted with the train set.

Listing 7 Example of classification using an instance of `DecisionTreeClassifier`.

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(x_train, y_train)
y_predicted = classifier.predict(x_test)
```

It is possible to calculate classification metrics for validation of the results by applying scikit-learn functions -
`classification_report` generates a full report that includes, for example, the support (number of predictions), precision, recall, and F1-score for each class, and the overall accuracy (see Listing 8).

To evaluate a model, we can apply cross-validation through `cross_val_score`. It returns the accuracy for each fold of the k-fold cross-validation, meaning that it is possible to calculate the average accuracy and standard deviation of a model using these values. Listing 9 shows an example of usage of this method. Note that it is possible to input the number of folds to the parameter `cv`, which is 5 by default.

Listing 8 Example of a report generated with `classification_report`.

```
>>> from sklearn.metrics import classification_report
>>> classification_report(y_test, y_predicted)
              precision    recall  f1-score   support

         yes       0.52      0.48      0.50       406
          no       0.52      0.56      0.54       409

    accuracy                           0.52       815
   macro avg       0.52      0.52      0.52       815
weighted avg       0.52      0.52      0.52       815
```

Listing 9 Usage of `cross_val_score` to cross-validate a model.

```
>>> from sklearn.model_selection import cross_val_score
>>> scores = cross_val_score(classifier, x_train, y_train, cv=5)
>>> print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
Accuracy: 0.53 (+/- 0.03)
```

2.6 Bibliographic Revision

In this section, we will briefly present the origins of the text analysis tool LIWC in Section 2.6.1. Then, in Section 2.6.2, we will present the state of the art in personality recognition from text through the enumeration of related works.

2.6.1 Linguistic Inquiry and Word Count

In 1996, Pennebaker and Francis tried to prove and explain how "writing about upsetting experiences can improve physical health" [35]. The study is based on cognitive factors known to be true from previous investigations: the writing process can make a person recall and identify more elements of a traumatic experience, more consciously and effortlessly over time, and that the act of writing or talking about the event can alter the form it is represented in memory.

The researchers recruited 72 students from Pennebaker's introductory psychology course. Those students were tasked with a writing assignment where they had to write continuously for 20 minutes, in three consecutive days, about their "very deepest thoughts and feelings about coming to college". For the analysis, a program called Linguistic Inquiry and Word Count, or LIWC, was developed, which is described in Section 2.4.3. The LIWC list of words was compiled from an array of sources, such as dictionaries and questionnaires, and each word was manually categorized by a group of three judges.

The authors concluded that "new students who wrote about their deepest thoughts and

feelings about coming to college evidenced improved physical health and academic performance compared to control subjects who wrote about superficial topics" [35].

2.6.2 Personality Recognition from Text

In this section we will present several works on personality identification from text. The following state of the art works use LIWC's features, which indicates that it is a standard for the proposed task. Also, all the works followingly presented are based on the Big Five personality traits model.

Mairesse et al., believing that recognizing personality from linguistic cues might be possible, created and described a selection of features that would become one of the most complete and used sets in related works to which they called *Mairesse baseline* features [49]. The authors gathered the collection of features based on previous results from psychological investigation on a relationship between linguistic and personality traits. Those features can be split into three categories:

- Context and syntax, composed by the frequency counts of 88 word categories from the LIWC program, and 14 additional features from the MRC Psycholinguistic database;
- Utterance type, categorizing each sentence into one of four labels: command, prompt, question, and assertion;
- Prosody, categories with focus on the rhythm and tone of the speech, such as speech rate and voice pitch;

The authors reported an average accuracy of, approximately, 57% using a SVM classifier.

Mohammad et al. investigated whether fine emotion categories are valid indicators of personality [12]. For that purpose, they made use of the Mairesse baseline features, the NRC Hashtag Lexicon, which contains the associations of 585 different emotions to words, aggregated from posts on Twitter social media platform, and the NRC Emotion Lexicon, which is described in Section 2.4.5. Furthermore, they employed some more lexicons for other linguistic dimensions, such as lexicons that correlate words to their evaluativeness, activity, and potency, and a dictionary that indicates a word's concept specificity. They followed a machine learning approach, using SVM classifiers, and tried different combinations of the previous groups of features. The authors corroborated that there is an evident relation between emotions and personality through the results. Those results indicated higher scores in classifiers that used the NRC Hashtag Lexicon, with an average accuracy of about 56%, therefore they concluded that the relations between the plethora of fine emotions and words are statistically significant.

Poria et al. [50] use a similar approach to the articles presented above, experimenting with a different set of features that consists of:

- LIWC features;
- MRC features - described in Section 2.4.6, these are extracted from the Medical Research Council database of psycholinguistic categories;
- EmoSenticSpace features, which is a blend of ConceptNet (described in Section 2.4.8) and EmoSenticNet (described in Section 2.4.7) made by the authors, and it was used to retrieve the polarity scores of each concept of the analyzed texts;
- Negation treatment, reversing the polarity score of the concept that followed the negation marker.

The classification they have done shows great performance, with an average accuracy of approximately 64% using a Sequential Minimal Optimization (SMO) classifier. The authors concluded that the use of links between concepts, and affective and sentiment information on the personality prediction problem manifests better outcomes than the algorithms solely based on linguistic features.

The work of Tighe et al. [51] focuses on the reduction of the LIWC2007 feature set, i.e. trying to remove non-relevant features in order to have the model fit the training data. From the 80 LIWC features, the authors removed these with zero information gain for each of the Big Five traits; for each personality trait, the categories that showed information gain are the following:

- Openness: Dictionary Words, Work, First Person Singular Pronoun, Second Person Pronoun, Home, Time, Relativity, Swear Words, Friends, All Punctuation, Motion, Religion, Parentheses, Articles, Commas, Regular Verbs, Personal Pronouns, Cognitive Processes, Function Words, Words with more than 6 letters, Question Marks, Sexuality, Quotation Marks, Anger;
- Conscientiousness: Swear words, Anger, Negative Emotion, Dictionary Words, Apostrophes, Function Words, Prepositions, Exclamation Marks;
- Extraversion: Articles, Personal Pronouns, Sexuality, Conjunctions;
- Agreeableness: Anger, Swear Words, Negative Emotion, Family, Dictionary Words;
- Neuroticism: Negative Emotion, Sadness, First Person Singular Pronoun, Anxiety, Personal Pronoun, Dictionary Words, Total Pronouns, Negations, Leisure.

To calculate the information gain for one feature, they used two methods: (i) calculate the information gain of each of LIWC's categories; (ii) identify patterns in the data using

principal component analysis. The authors experimented with two implementations of SVM (*LibSVM* and *SMO*) and one Linear Logistic Regression algorithm (*SimpleLogistic*). The results a slight improvement in the classification when using the feature-reduced data sets and the SVM implementations, with an average accuracy of around 57.9% and an improvement of approximately 0.9% over the average accuracy of the classification using all features, and a massive reduction of the LIWC dataset's size (over 70% of the original size).

2.7 Summary

In this chapter, we covered the cornerstone concepts for this work, from more theoretical topics to more technical subjects. In Section 2.2 we introduced the concept and the existing definitions of personality, elaborated on the trait approach to personality, and described the relationship between psychology and emotion.

In Section 2.3, we enumerated the several steps that compose the process of KDD. First, we presented seven common pre-processing techniques: aggregation, sampling, dimensionality reduction, feature subset selection, feature creation, discretization and binarization, and variable transformation. Then, we briefly described the data mining step and three common data mining methods: classification, regression, and clustering. Next, we presented several challenges to the KDD application, namely high dimensionality, overfitting, changing data and knowledge, missing and noisy data, complex relationships between fields, understandability of patterns, and user interaction and prior knowledge. Finally, we enumerate tasks that are commonly part of the postprocessing phase, specifically knowledge filtering, interpretation and explanation, knowledge integration, evaluation, and validation of results.

Section 2.4 begins with an introduction to linguistics by describing the concepts of syntax, morphology, lexicon, semantics, and pragmatics. Then, we detail the following NLP techniques and tools: the word embedding method, the Linguistic Inquiry and Word Count text analysis tool, WordNet, the NRC Emotion Lexicon, the MRC Psycholinguistic Database, EmoSenticNet, ConceptNet, and the Natural Language Toolkit suite.

In Section 2.5, we introduce four distinct machine learning algorithms for classification that we used during this work: the Naive Bayes classifier, Decision Tree learning, Support-Vector Machine, and Random Forest. In this section, we also described Scikit-learn, which is a machine learning module for Python.

In Section 2.6, we presented the work behind the creation of LIWC and the state of the art in personality recognition. All related work described use LIWC's features for personality identification, designating it as a standard data set for feature extraction, and the common

choice for personality modeling is the Big Five personality trait model.

Chapter 3

Methodology

3.1 Introduction

This chapter elaborates on the implementation of the approach proposed in 1.3. Section 3.2 clarifies the data preprocessing step of our KDD task. Later, in Section 3.3, we explain the vocabulary expansion procedure. At last, we expose how we classify texts in order to recognize personality, and how we evaluate the classification ratings in Section 3.3.

3.2 Preprocessing

The preprocessing stage consists of several steps with the goal of preparing the input data to be used in the classification phase.

First, we prepared our dataset of 2468 stream-of-consciousness essays, which originates from Pennebaker and Francis' work described in Section 2.6.1. An annotated excerpt of a sample can be seen in Table 3.1, and some integral essays can be read in Appendix B.1.

Table 3.1: Annotated sample from the stream-of-consciousness essays dataset.

ID	TEXT	cEXT	cNEU	cAGR	cCON	cOPN
1997_504851.txt	Well, right now I just woke up from a mid-day nap. It's sort of weird, but ever since I moved to Texas, I have had problems concentrating on things. (...)	n	y	y	n	y

Each essay is annotated with a value for each of the Big Five's factors for the essay author's personality. In the version of the dataset we possess, we found one essay that had to be removed due to faulty content ("Err:508"), so we ended up with an aggregate of 2,467 stream-of-consciousness essays.

The first process we use to transform the texts is tokenization. Every essay is converted to lower case, and then a tokenizer from NLTK is used to split the text into tokens. The tokenizer uses a custom regular expression to catch tokens with one or fewer apostrophes or hyphens (e.g., "I'm", "e-mail"). This process produces objects such as the one in Listing

10.

Listing 10 Object created after tokenizing the stream-of-consciousness essays.

```
{
  "id": "1997_504851.txt",
  "text": "well, right now i just woke up from a mid-day nap.
           it's sort of weird, but ever since i moved to texas,
           i have had problems concentrating on things.",
  "cext": "n",
  "cneu": "y",
  "cagr": "y",
  "ccon": "n",
  "copn": "y",
  "tokens": [
    "well", "right", "now", "i", "just", "woke", "up", "from",
    "a", "mid-day", "nap", "it's", "sort", "of", "weird", "but",
    "ever", "since", "i", "moved", "to", "texas", "i", "have",
    "had", "problems", "concentrating", "on", "things"
  ]
}
```

The next step consists of trying to find matches in LIWC's dictionary for each token of each essay. If a match is found, i.e. the token exists in the dictionary, then we increment the counter for the categories associated with that word. For example, the word "living" is annotated with the categories "verb" (20 - "Verbs"), "bio" (70 - "Biological Processes") and "health" (72 - "Health"), so we increment the counter of each of those categories for the essay in which the word appears. There are some cases that need special processing: the treatment of keys with wildcards, the occurrence of the pair "kind of", and the happening of sequences with the word "like".

The process is sequential, meaning that we try to match words sequentially as they appear in the original text. When matching a word with LIWC's dictionary, the algorithm searches for the 10 closest keys from the dictionary. If an exact match is found, the word is returned. If an exact match is not found, we search through the 10 possibilities for a key that includes the wildcard character "*" and partially match our token with the key; if the token matches the key up to the wildcard, the key is returned.

While looping through the tokens, if the token to match is "kind" we look to the next token to see if it is the term "of". The expression "kind of" must be treated as a single word since it is a key in LIWC's dictionary.

The problem with the word "like" is similar to the previous case, although the word sequence is reversed. The dictionary has several branches for groups of words that include the term "like":

- to like
- (i) like*
- (you) like*
- (we) like*
- (they) like*
- (do) like
- (don't) like
- (did) like
- (didn't) like
- (will) like
- (won't) like
- (does) like
- (doesn't) like
- (did not) like
- (will not) like
- (do not) like
- (does not) like
- (would not) like
- (should not) like
- (could not) like
- (53) like*

To handle these situations, while looping through the essay's tokens, we look not only to the current word but also to the next two tokens and, if any of the next two tokens is the word "like", try to match the sequence of words with these special keys. Note the last key of the previous list with the number 53 in it - it means that any word associated with the category number 53 ("discrep" - "Discrepancies") followed by any word that matches the pattern "like*" is a match for this key (e.g., "hopefully likes").

At the end of this step, we have a list of objects, each containing the essay identifier, the counter for each category of LIWC, and the total number of words matched with the dictionary. Listing 11 shows an example of an object of this type.

Listing 11 Object resulting from the dictionary matching.

```
{
  "id": "1997_504851.txt",
  "function": 409,
  "pronoun": 123,
  "ppron": 81,
  // Omitted the remaining 70 categories
  "dic": 628
}
```

The last preprocessing step converts our algorithm's analysis into an output similar to the LIWC's text processing module's output. First, we count the total number of words in each essay. Using this value, the category counts are transformed into percentages of the total of words; for instance, for an essay with 661 words from which 409 are function words, the "function" category would be represented by the rounded value of 61.9%.

It also calculates three additional features: number of words per sentence, percentage of words with more than six letters, and percentage of essay words found in the dictionary.

There are four summary variables in LIWC's output that we were not able to determine

Listing 12 Output for an essay at the end of the preprocessing phase.

```
{
    "id": "1997_504851.txt",
    "function": 61.87594553706505,
    "pronoun": 18.608169440242058,
    "ppron": 12.254160363086234,
    // Omitted the remaining 70 categories
    "dic": 95.00756429652043,
    "wordcount": 661,
    "wps": 18.885714285714286,
    "sixltr": 9.379727685325264,
    "cext": 0,
    "cneu": 1,
    "cagr": 1,
    "ccon": 0,
    "copn": 1
}
```

as they use proprietary algorithms derived from previous research of the authors. Refer to Section 2.4.3 for more information.

The final output for an essay is given by a dictionary that includes the essay identifier, the percentage values for each category, the percentage of words found in the dictionary, the word count, the number of words per sentence, the percentage of words with more than six letters, and the Big Five factors' values transformed from 'n' and 'y' to 0 and 1, respectively.

3.3 Vocabulary expansion

As an effort to achieve better results, we expanded LIWC's vocabulary to be able to categorize a larger range of tokens. Two different methods were used to expand the dictionary: WordNet's sets of cognitive synonyms (synsets), and word embeddings.

The first method consists of finding synonyms using WordNet's collection of synsets, which are described in more detail in Section 2.4.4. This method takes into account the part of speech of each word, i.e. whether the word is a noun, verb, adjective, or adverb, except for LIWC's keys with wildcard characters. The LIWC dictionary allows us to know which part of speech a word belongs to through the category annotations; the word is a verb if annotated with the number 20, adjective if 21, adverb if 13, and all others are considered nouns.

NLTK provides an interface for WordNet's methods, which allows to easily query synsets for a specific word. After getting the synsets, we check whether these words exist in LIWC's

dictionary and, if not, add them to it with the same categories as the ones the original word has. Listing 3 exhibits examples of usage of this interface.

We only add the first synset out of all synsets found for a word, for it represents the most common sense for that word. We noticed that occasionally some senses were being wrongfully considered, e.g. the word "zen", which is associated to the religion subcategory in LIWC, can be interpreted in the sense of a drug ("street name for lysergic acid diethylamide") and, for that, the word "acid" was being added to the base dictionary with an association to religion.

Examples of successful cases of vocabulary extension through this method are:

- "modification" added as a synonym of "change", with the categories "cognitive processes", "causal", "relativity" and "motion";
- "valiance" added as a synonym of "heroism", associated to "affect" and "positive emotions";
- "remuneration" added as a synonym of "wage", which is connected to "work" and "money".

The expansion using word embeddings is done by finding similar words for a target word. Since words are mapped to vectors of real numbers we can use cosine similarity to measure the similarity between two words and, consequently, find words that are similar to a target word. Our algorithm goes through the dictionary and searches for the most similar vectors above a specific threshold for each word of the dictionary - the results that satisfy the threshold condition are added to the dictionary with the same categories as the query word.

The word embeddings data set we used is a subset of word2vec's data set (described in Section 2.4.2) that excludes terms with low frequency in a selected corpus, like strings that are not English words.

3.4 Classification and Evaluation

The classification step consists of trying to derive models from training data and evaluating those using test data, intensively using Scikit-learn for those purposes.

To split the data set in training and testing data set and validate the created models we use the cross-validation method, which is introduced in Section 2.3.5. We decided to run

the k-fold cross-validation method with $k = 10$. Aiming for more accurate results, we also shuffle the data set prior to the split to reduce the eventuality of overfitting problems.

The data set is classified using four different machine learning algorithms: *Multinomial Naive Bayes*, *Decision Tree*, *Support-Vector Machine*, and *Random Forest*. These learning models are detailed in Section 2.5. The classifiers used the default hyperparameters established in Scikit-learn.

The decision behind the Multinomial Naive Bayes and Support-Vector Machine is supported by our bibliographic revision, detailed in Section 2.6. Related works have reported satisfactory results using Naive Bayes and Support-Vector Machine classifiers, therefore we used them to create a ground for comparison. The Decision Tree and Random Forest algorithms are suitable for this classification task since they are simple to understand, require little data preparation because they accept not only quantitative but also qualitative data, have in-built feature selection, and exhibit good performance for data sets with a size similar to ours (2468 samples). In the experiments made so far, we avoid the use of deep learning techniques because our data set is not large enough to achieve a good model.

After fitting the models with the training sets and classifying the test sets, we evaluate the results by measuring some key indicators, namely the *accuracy*, the *precision*, the *recall*, and the *F1* score for each model. The referred metrics are explained in Section 2.3.5.

The classification process was executed a total of four times, one for each dictionary input. The essays were classified using the original LIWC dictionary, the dictionary expanded using WordNet, the dictionary expanded using word embeddings, and the baseline dictionary expanded using both methods.

3.5 Summary

Over this chapter, we described the steps that compose the algorithm we developed. First, we clarify how we preprocess the input essays, from the word tokenization to the dictionary matching. Then, we detailed our methodology of vocabulary expansion. Finally, we described the classification methods experimented, and how we evaluate the results obtained through the conventional measures.

Chapter 4

Results and Discussion

4.1 Introduction

This chapter presents the results obtained through the methodology described in Chapter 3, and the author's critical assessment. The results are presented in Section 4.2, and the same are discussed in Section 4.3.

4.2 Results

The classification step produced four different groups of results, one for each data set used: the base LIWC's dictionary, the dictionary expanded using WordNet's synsets, the dictionary extended using the word embeddings, and the dictionary with both expansion outcomes.

4.2.1 Baseline data set

Tables 4.1 to 4.4 show the results obtained by the various classifiers using the default data set. The Random Forest classifier shows the best outcomes with an average F1-score of approximately 56.2%. The worst classification metrics resulted from the Support-Vector Machine classifier which exhibits an average F1-score of about 41.6%.

The algorithm was able to find about 93.2% of the essays' words in the dictionary's vocabulary.

Table 4.1: Results for the Naive Bayes classifier using the **baseline** data set.

Trait	**Accuracy**	**Precision**	**Recall**	**F1-score**
Openness	0.533 ± 0.037	0.542 ± 0.041	0.538 ± 0.036	0.523 ± 0.040
Conscientiousness	0.530 ± 0.024	0.536 ± 0.032	0.526 ± 0.024	0.496 ± 0.027
Extraversion	0.504 ± 0.025	0.503 ± 0.026	0.502 ± 0.024	0.499 ± 0.023
Agreeableness	0.540 ± 0.030	0.540 ± 0.034	0.534 ± 0.029	0.527 ± 0.030
Neuroticism	0.528 ± 0.033	0.528 ± 0.034	0.528 ± 0.033	0.526 ± 0.033
Average	0.527	0.530	0.526	0.514

Table 4.2: Results for the Decision Tree classifier using the **baseline** data set.

Trait	Accuracy	Precision	Recall	F1-score
Openness	0.539 ± 0.026	0.539 ± 0.026	0.538 ± 0.026	0.538 ± 0.026
Conscientiousness	0.499 ± 0.034	0.499 ± 0.034	0.499 ± 0.034	0.499 ± 0.034
Extraversion	0.510 ± 0.019	0.509 ± 0.019	0.509 ± 0.019	0.508 ± 0.019
Agreeableness	0.499 ± 0.022	0.496 ± 0.023	0.496 ± 0.023	0.496 ± 0.023
Neuroticism	0.517 ± 0.024	0.517 ± 0.024	0.517 ± 0.024	0.517 ± 0.023
Average	0.513	0.512	0.512	0.512

Table 4.3: Results for the Support-Vector Machine classifier using the **baseline** data set.

Trait	Accuracy	Precision	Recall	F1-score
Openness	0.524 ± 0.018	0.529 ± 0.040	0.512 ± 0.018	0.440 ± 0.022
Conscientiousness	0.507 ± 0.013	0.510 ± 0.034	0.501 ± 0.013	0.423 ± 0.042
Extraversion	0.509 ± 0.012	0.437 ± 0.146	0.494 ± 0.010	0.376 ± 0.034
Agreeableness	0.530 ± 0.002	0.290 ± 0.075	0.499 ± 0.002	0.347 ± 0.002
Neuroticism	0.511 ± 0.052	0.514 ± 0.058	0.511 ± 0.052	0.493 ± 0.052
Average	0.516	0.456	0.503	0.416

Table 4.4: Results for the Random Forest classifier using the **baseline** data set.

Trait	Accuracy	Precision	Recall	F1-score
Openness	0.602 ± 0.029	0.602 ± 0.029	0.602 ± 0.029	0.601 ± 0.029
Conscientiousness	0.546 ± 0.025	0.546 ± 0.025	0.546 ± 0.025	0.546 ± 0.025
Extraversion	0.556 ± 0.020	0.555 ± 0.020	0.554 ± 0.020	0.553 ± 0.019
Agreeableness	0.553 ± 0.031	0.549 ± 0.032	0.547 ± 0.031	0.543 ± 0.032
Neuroticism	0.568 ± 0.036	0.568 ± 0.036	0.568 ± 0.036	0.567 ± 0.036
Average	0.565	0.564	0.563	0.562

Table 4.5: Comparison of the F1-scores obtained by the classifiers for all traits using the **baseline** data set.

	Naive Bayes	Support-Vector Machine	Decision Tree	Random Forest
Openness	0.523	0.440	0.538	0.601
Conscientiousness	0.496	0.426	0.499	0.546
Extraversion	0.499	0.376	0.508	0.553
Agreeableness	0.527	0.347	0.496	0.543
Neuroticism	0.526	0.493	0.517	0.567
Average	0.514	0.416	0.512	0.562

4.2.2 WordNet extension

Tables 4.6 to 4.9 show the results attained by the classifiers using the baseline dictionary with the WordNet's synsets extension. The Random Forest classifier presents the best results out of the four classifiers, with an average F1-score of approximately 55.6%. The Support-Vector machine classifier shows the worst metrics at an average F1-score of about 41.6%.

The WordNet extension added 1186 new words to the LIWC dictionary, and we got an average percentage of essay words found in the dictionary of, approximately, 93.6%.

Table 4.6: Results for the Naive Bayes classifier using the **WordNet-extended** data set.

Trait	Accuracy	Precision	Recall	F1-score
Openness	0.532 ± 0.031	0.542 ± 0.036	0.537 ± 0.031	0.519 ± 0.034
Conscientiousness	0.519 ± 0.023	0.524 ± 0.030	0.515 ± 0.022	0.485 ± 0.017
Extraversion	0.512 ± 0.042	0.507 ± 0.046	0.510 ± 0.043	0.501 ± 0.053
Agreeableness	0.531 ± 0.035	0.526 ± 0.037	0.523 ± 0.034	0.513 ± 0.040
Neuroticism	0.530 ± 0.032	0.531 ± 0.033	0.530 ± 0.032	0.528 ± 0.032
Average	0.525	0.526	0.523	0.509

Table 4.7: Results for the Decision Tree classifier using the **WordNet-extended** data set.

Trait	Accuracy	Precision	Recall	F1-score
Openness	0.533 ± 0.036	0.533 ± 0.037	0.533 ± 0.037	0.533 ± 0.037
Conscientiousness	0.504 ± 0.036	0.504 ± 0.036	0.504 ± 0.036	0.504 ± 0.036
Extraversion	0.514 ± 0.029	0.513 ± 0.029	0.513 ± 0.029	0.513 ± 0.029
Agreeableness	0.520 ± 0.027	0.519 ± 0.028	0.519 ± 0.027	0.518 ± 0.027
Neuroticism	0.509 ± 0.026	0.509 ± 0.026	0.509 ± 0.026	0.509 ± 0.025
Average	0.516	0.516	0.516	0.515

Table 4.8: Results for the Support-Vector Machine classifier using the **WordNet-extended** data set.

Trait	Accuracy	Precision	Recall	F1-score
Openness	0.526 ± 0.018	0.533 ± 0.041	0.515 ± 0.018	0.443 ± 0.023
Conscientiousness	0.513 ± 0.020	0.494 ± 0.089	0.507 ± 0.020	0.430 ± 0.043
Extraversion	0.514 ± 0.019	0.520 ± 0.173	0.498 ± 0.018	0.365 ± 0.028
Agreeableness	0.530 ± 0.002	0.265 ± 0.001	0.499 ± 0.002	0.346 ± 0.001
Neuroticism	0.514 ± 0.025	0.517 ± 0.029	0.514 ± 0.025	0.496 ± 0.023
Average	0.519	0.466	0.507	0.416

Table 4.9: Results for the Random Forest classifier using the **WordNet-extended** data set.

Trait	Accuracy	Precision	Recall	F1-score
Openness	0.591 ± 0.025	0.591 ± 0.026	0.590 ± 0.026	0.590 ± 0.026
Conscientiousness	0.536 ± 0.029	0.536 ± 0.029	0.535 ± 0.029	0.534 ± 0.029
Extraversion	0.564 ± 0.023	0.563 ± 0.023	0.563 ± 0.022	0.562 ± 0.022
Agreeableness	0.536 ± 0.032	0.531 ± 0.034	0.530 ± 0.032	0.527 ± 0.033
Neuroticism	0.567 ± 0.014	0.567 ± 0.014	0.567 ± 0.014	0.566 ± 0.014
Average	0.559	0.558	0.557	0.556

Table 4.10: Comparison of the F1-scores obtained by the classifiers for all traits using the **WordNet-extended** data set.

	Naive Bayes	Support-Vector Machine	Decision Tree	Random Forest
Openness	0.519	0.443	0.533	0.590
Conscientiousness	0.485	0.430	0.504	0.534
Extraversion	0.501	0.365	0.513	0.562
Agreeableness	0.513	0.346	0.518	0.527
Neuroticism	0.528	0.496	0.509	0.566
Average	0.509	0.416	0.515	0.556

4.2.3 Word embeddings extension

Tables 4.11 to 4.14 show the results for the experiment with the baseline data set extended through the usage of word embeddings. The Random Forest classifier indicates an average F1-score of approximately 55.6%, which is the best result obtained for this extension. The Support-Vector machine classifier exhibits the poorest metrics at an average F1-score of about 41.6%.

Using word embeddings to enlargen the original vocabulary introduced 815 new words, and the mean percentage of essay words captured was, roughly, 93.4%.

Table 4.11: Results for the Naive Bayes classifier using the **embeddings-extended** data set.

Trait	**Accuracy**	**Precision**	**Recall**	**F1-score**
Openness	0.531 ± 0.035	0.543 ± 0.044	0.536 ± 0.036	0.518 ± 0.035
Conscientiousness	0.531 ± 0.031	0.538 ± 0.040	0.528 ± 0.030	0.499 ± 0.032
Extraversion	0.501 ± 0.030	0.499 ± 0.031	0.499 ± 0.030	0.492 ± 0.030
Agreeableness	0.542 ± 0.056	0.539 ± 0.058	0.537 ± 0.056	0.534 ± 0.056
Neuroticism	0.535 ± 0.029	0.536 ± 0.029	0.535 ± 0.029	0.532 ± 0.030
Average	0.528	0.531	0.527	0.515

Table 4.12: Results for the Decision Tree classifier using the **embeddings-extended** data set.

Trait	**Accuracy**	**Precision**	**Recall**	**F1-score**
Openness	0.547 ± 0.030	0.547 ± 0.030	0.547 ± 0.030	0.546 ± 0.030
Conscientiousness	0.511 ± 0.029	0.511 ± 0.029	0.511 ± 0.029	0.511 ± 0.029
Extraversion	0.513 ± 0.033	0.512 ± 0.033	0.512 ± 0.033	0.512 ± 0.033
Agreeableness	0.508 ± 0.034	0.507 ± 0.034	0.507 ± 0.034	0.507 ± 0.034
Neuroticism	0.512 ± 0.021	0.513 ± 0.022	0.512 ± 0.022	0.512 ± 0.021
Average	0.518	0.518	0.518	0.518

Table 4.13: Results for the Support-Vector Machine classifier using the **embeddings-extended** data set.

Trait	**Accuracy**	**Precision**	**Recall**	**F1-score**
Openness	0.525 ± 0.020	0.529 ± 0.043	0.513 ± 0.020	0.440 ± 0.025
Conscientiousness	0.508 ± 0.018	0.477 ± 0.094	0.503 ± 0.017	0.426 ± 0.046
Extraversion	0.516 ± 0.004	0.406 ± 0.165	0.500 ± 0.004	0.367 ± 0.037
Agreeableness	0.530 ± 0.003	0.265 ± 0.001	0.499 ± 0.002	0.346 ± 0.001
Neuroticism	0.517 ± 0.028	0.520 ± 0.033	0.517 ± 0.028	0.500 ± 0.029
Average	0.519	0.439	0.506	0.416

Table 4.14: Results for the Random Forest classifier using the **embeddings-extended** data set.

Trait	**Accuracy**	**Precision**	**Recall**	**F1-score**
Openness	0.587 ± 0.022	0.586 ± 0.022	0.586 ± 0.022	0.586 ± 0.022
Conscientiousness	0.549 ± 0.017	0.549 ± 0.017	0.549 ± 0.017	0.548 ± 0.017
Extraversion	0.537 ± 0.021	0.535 ± 0.021	0.535 ± 0.021	0.534 ± 0.021
Agreeableness	0.566 ± 0.038	0.563 ± 0.040	0.558 ± 0.036	0.554 ± 0.036
Neuroticism	0.559 ± 0.026	0.559 ± 0.026	0.559 ± 0.026	0.558 ± 0.026
Average	0.560	0.558	0.557	0.556

Table 4.15: Comparison of the F1-scores obtained by the classifiers for all traits using the **embeddings-extended** data set.

	Naive Bayes	Support-Vector Machine	Decision Tree	Random Forest
Openness	0.519	0.443	0.533	0.590
Conscientiousness	0.485	0.430	0.504	0.534
Extraversion	0.501	0.365	0.513	0.562
Agreeableness	0.513	0.346	0.518	0.527
Neuroticism	0.528	0.496	0.509	0.566
Average	0.509	0.416	0.515	0.556

4.2.4 WordNet and word embeddings extensions

Tables 4.16 to 4.19 show the results of the classification using the baseline data set extended through both WordNet and word embeddings methods. The Random Forest classifier presents the leading values with an average F1-score of approximately 55.4%. The Support-Vector machine classifier manifests the most unfavorable calculations with an average F1-score of about 41.8%.

Both extension methods account for a total of 2001 added words, and the average percentage of words matched for this experiment is around 93.8%.

Table 4.16: Results for the Naive Bayes classifier using **both WordNet and Embeddings extension** data set.

Trait	Accuracy	Precision	Recall	F1-score
Openness	0.530 ± 0.019	0.545 ± 0.034	0.535 ± 0.020	0.516 ± 0.017
Conscientiousness	0.528 ± 0.021	0.535 ± 0.035	0.524 ± 0.021	0.495 ± 0.027
Extraversion	0.513 ± 0.036	0.511 ± 0.039	0.511 ± 0.038	0.503 ± 0.040
Agreeableness	0.531 ± 0.019	0.526 ± 0.019	0.524 ± 0.019	0.514 ± 0.027
Neuroticism	0.536 ± 0.026	0.537 ± 0.027	0.536 ± 0.026	0.533 ± 0.027
Average	0.528	0.531	0.526	0.512

Table 4.17: Results for the Decision Tree classifier using **both WordNet and Embeddings extension** data set.

Trait	Accuracy	Precision	Recall	F1-score
Openness	0.532 ± 0.033	0.532 ± 0.033	0.531 ± 0.033	0.531 ± 0.033
Conscientiousness	0.530 ± 0.032	0.530 ± 0.032	0.530 ± 0.032	0.529 ± 0.032
Extraversion	0.497 ± 0.036	0.497 ± 0.036	0.496 ± 0.036	0.495 ± 0.035
Agreeableness	0.532 ± 0.030	0.530 ± 0.030	0.530 ± 0.030	0.530 ± 0.031
Neuroticism	0.511 ± 0.027	0.511 ± 0.027	0.511 ± 0.027	0.510 ± 0.027
Average	0.520	0.520	0.520	0.519

Table 4.18: Results for the Support-Vector Machine classifier using **both WordNet and Embeddings extension** data set.

Trait	Accuracy	Precision	Recall	F1-score
Openness	0.527 ± 0.015	0.539 ± 0.038	0.516 ± 0.015	0.447 ± 0.015
Conscientiousness	0.504 ± 0.023	0.500 ± 0.041	0.499 ± 0.023	0.427 ± 0.050
Extraversion	0.514 ± 0.009	0.510 ± 0.165	0.499 ± 0.008	0.368 ± 0.029
Agreeableness	0.530 ± 0.002	0.265 ± 0.001	0.499 ± 0.002	0.346 ± 0.001
Neuroticism	0.521 ± 0.038	0.523 ± 0.044	0.521 ± 0.038	0.503 ± 0.043
Average	0.519	0.467	0.507	0.418

Table 4.19: Results for the Random Forest classifier using **both WordNet and Embeddings extension** data set.

Trait	Accuracy	Precision	Recall	F1-score
Openness	0.581 ± 0.022	0.580 ± 0.022	0.580 ± 0.022	0.580 ± 0.022
Conscientiousness	0.545 ± 0.025	0.545 ± 0.025	0.545 ± 0.025	0.544 ± 0.025
Extraversion	0.540 ± 0.031	0.538 ± 0.032	0.537 ± 0.032	0.537 ± 0.032
Agreeableness	0.548 ± 0.025	0.544 ± 0.026	0.542 ± 0.025	0.538 ± 0.026
Neuroticism	0.569 ± 0.014	0.569 ± 0.014	0.569 ± 0.014	0.569 ± 0.014
Average	0.557	0.555	0.555	0.554

Table 4.20: Comparison of the F1-scores obtained by the classifiers for all traits using **both WordNet and Embeddings extension** data set.

	Naive Bayes	Support-Vector Machine	Decision Tree	Random Forest
Openness	0.516	0.447	0.531	0.580
Conscientiousness	0.495	0.427	0.529	0.544
Extraversion	0.503	0.368	0.495	0.537
Agreeableness	0.514	0.346	0.530	0.538
Neuroticism	0.533	0.503	0.510	0.569
Average	0.512	0.418	0.519	0.554

4.3 Discussion

Table 4.21 shows the F1-score metrics for each of the Big Five traits - openness (O), conscientiousness (C), extraversion (E), agreeableness (A), and neuroticism (N), in this order - and their averages, which are the result of the classification process using the four different classifiers (Naive Bayes, Decision Tree, Support-Vector Machine, and Random Forest) for each of the four different sets of data (LIWC's dictionary (Base), the dictionary extended with WordNet (WN) synsets (WN), the dictionary extended with word embeddings (WE), and the dictionary extended with both methods (Both)).

We noticed that the Support-Vector Machine classifier stands out for its poor results, and as such these will not be considered for comparison throughout the rest of this section. We will evaluate what went wrong later on in this section.

Regarding the dictionary matches, we can observe that there were no significant differ-

Table 4.21: Comparison of the F1-score values for the Big Five traits, obtained by the various classifiers on the four experiments.

	Naive Bayes				Decision Tree				Support-Vector Machine				Random Forest			
	Base	**WN**	**WE**	**Both**	**Base**	**WN**	**WE**	**Both**	**Base**	**WN**	**WE**	**Both**	**Base**	**WN**	**WE**	**Both**
O	0.523	0.519	0.518	0.516	0.538	0.533	0.546	0.531	0.440	0.443	0.440	0.447	0.601	0.590	0.586	0.580
C	0.496	0.485	0.499	0.495	0.499	0.504	0.511	0.529	0.423	0.430	0.426	0.427	0.546	0.534	0.548	0.544
E	0.499	0.501	0.492	0.503	0.508	0.513	0.512	0.495	0.376	0.365	0.367	0.468	0.553	0.562	0.534	0.537
A	0.527	0.513	0.534	0.514	0.496	0.518	0.507	0.530	0.347	0.346	0.346	0.346	0.543	0.527	0.554	0.538
N	0.526	0.528	0.532	0.533	0.517	0.509	0.512	0.510	0.493	0.496	0.500	0.503	0.567	0.566	0.558	0.569
Avg	0.514	0.509	0.515	0.512	0.512	0.515	0.518	0.519	0.416	0.416	0.416	0.418	0.562	0.556	0.556	0.554

ences introduced by the vocabulary extensions. The extension using both described methods exhibits an increase of 0.6% in the percentage of words found in the dictionary over the base vocabulary, which means that the LIWC dictionary reaches a number of English words large enough that the performed additions may not bring value.

The Naive Bayes classifier presents an average F1-score of 51.4% when using the base data set. For the two separate LIWC expansions - WordNet synsets, and word embeddings - it shows an average F1-score of, respectively, 50.9% and 51.5%, which translates to a gain of 0.1% when word embeddings are added, and a loss of 0.5% when WordNet synsets are included. Training the model with LIWC's dictionary with both expansions results in an average F1-score of 51.2%, which is a 0.2% loss in comparison to the base data set.

The Decision Tree classifier manifests an improvement with the successive base data set extensions. The base data set experiment resulted in an average F1-score of 51.2%, and it is increased to 51.5%, 51.8%, and 51.9% for the WordNet, word embeddings, and both methods experiments, respectively.

The Random Forest classifier exhibits a behavior contrary to the Decision Tree; the results decrease after the addition of new vocabulary to the initial data group. The average F1-scores obtained are 56.2%, 55.6%, 55.6%, and 55.4% for the classifications using the baseline data set, WordNet extension, word embeddings expansion, and both methods, respectively.

The average F1-score changes that occur when the data set is enlargened are not considered significant, as the largest recorded differences are a loss of 0.8% in the random forest marks and a gain of 0.7% in the decision tree ratings. However, if we consider each Big Five personality trait individually we are able to identify substantial variations.

The Decision Tree classifications show an improvement of 3.0% and 3.4%, respectively, for the conscientiousness and the agreeableness traits when comparing the baseline data set experiment with the assessment that uses both expansion methods. On the other hand, it presents a reduction of 1.3% for the extraversion trait. We assert that expanding LIWC's dictionary may not lead to a global information gain or loss, but can improve or decrease the classification of individual traits. It also depends on which classifier we are using since, for instance, benefits that can be seen in the decision tree ratings cannot be perceived in

Table 4.22: Comparison of the developed work's accuracy results with related works'.

	[49]	[12]	[50]	Present work (base)	Present work (both methods)
Openness	0.621	0.604	0.661	0.602	0.581
Conscientiousness	0.553	0.565	0.633	0.546	0.545
Extraversion	0.549	0.547	0.634	0.556	0.540
Agreeableness	0.558	0.540	0.615	0.553	0.548
Neuroticism	0.574	0.557	0.637	0.568	0.569
Average	0.571	0.563	0.636	0.565	0.557

the Naive Bayes'.

We chose three different works ([49], [12], [50]) on personality recognition that also use LIWC's features. These projects use SVMs and achieve good outcomes, which is not the case of our experiments. This may be due to not having fine-tuned the classifier's hyper-parameters. Also, the fact that they have better results may be a consequence of them having added features that we did not, e.g. in [49] the authors introduced features from the MRC database in addition to LIWC's features.

Both Naive Bayes and Decision Tree classifiers present results below the ones of existing works. However, the random forest classifier shows results similar to the ratings reported by two of the three related projects. Table 4.22 compares the results of the three referred articles with ours.

The accuracy values for our classification using the default LIWC vocabulary are higher than the reports in [12] for all traits except conscientiousness, and similar to the ratings of [49]. However, they are significantly below the results of [50], which shows that the referred work benefits from the added features.

As the present work is based on an unsupervised vocabulary extension, we do not have total control over what words are added to the original dictionary and, as such, we can introduce words that do not share the sense of the original word. We tried to address every case where words were being associated with wrong categories, such as the case described in 3.3, but eventually some cases might have been missed.

4.4 Summary

The results obtained by the classifiers show that there is not a significant difference, on average, when expanding LIWC's dictionary with the proposed methods. Although, distinct classifiers exhibit disparate information gains and losses for individual traits, which may mean that the vocabulary extensions may benefit the classication of individual traits.

The support-vector machine classifier presented the worst outcomes out of the four. This

is a peculiar finding as related works achieve the best results with SVMs. It may be due to the additional features used in the other projects, and/or to the classifier's hyperparameters configuration.

The random forest classifier revealed the best results, which are similar to the ones reported in the referred articles. However, every method of expansion produced decreases in scores, both in average and in individual trait values, which indicates that there was not any information gain through the vocabulary additions.

Chapter 5

Conclusions and Future Work

This final chapter presents the main conclusions of the work described, in the present dissertation in Section 5.1, and specifies potential points of improvement for the work we have done, in Section 5.2.

5.1 Main Conclusions

In the beginning of this dissertation we set out to investigate two points on personality recognition from written text, after detecting them as potential gaps in the reviewed literature, as follows:

1. Assess whether expanding LIWC's vocabulary, to catch a larger range of English words, is beneficial for the personality identification task from vocabulary;
2. Experiment with classifiers other than SVMs, which were the type of classifiers that produced best results in similar works, to try and find one that displays better ratings.

Regarding the first topic, the vocabulary expansion did not yield better results. The differences in the metrics from the classification using the LIWC dictionary alone are negligible when compared with the ratings from the classification using an extended dictionary. Also, the average percentage of matched words from the essays has increased, at most, by less than 1% using our vocabulary extensions, which means the LIWC dictionary's range is large enough that the expansions did not have a crucial impact.

As for the second point, we have found a machine learning algorithm that shows promising results: random forests. The results we present in this dissertation show that this classifier performs similarly to the models trained in related works. These related works also utilize the LIWC dictionary for feature extraction, however, they make use of additional data sets for the same effect (MRC Psycholinguistic Database, SenticNet, WNA, among others). For this reason, we believe the classification using random forests would exhibit favorable ratings if the feature extraction processed was improved.

Finally, we think we have condensed useful information about the state of the art on personality discovery from text and laid the groundwork for experiments of interest, which we suggest in Section 5.2.

5.2 Future Work

During the discussion presented in Section 4.3, we have identified various potential courses of action that one can implement to improve on our approach.

One suggestion would be to repeat the experiments described in this document using the same classifier algorithms but this time with tuned hyperparameters. The machine learning algorithms used in this work have configurable hyperparameters that control their behavior and will likely see their performance improved if the hyperparameters are refined.

Another possibility would be to experiment using the random forest classifier, since it gave the best results out of the four different algorithms, and additional data sets for feature extraction. The idea would be to, for example, take the feature extraction approach described in [50] - the one that yielded the best ratings - and classify the data set using a random forest classifier.

As shown in Section 2.2.3, personality is tightly linked to emotion and, as such, one possible experiment would be to use a word embeddings data set based on a corpus related to emotion and sentiment analysis.

Bibliography

[1] J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn, "The development and psychometric properties of LIWC2015," Tech. Rep., 2015. vii, xv, 1, 27, 28

[2] A. Vital, "5 personality traits - infographic," Nov 2018. [Online]. Available: https://blog.adioma.com/5-personality-traits-infographic/ xiii, 11

[3] "Emotion classification," Sep 2020. [Online]. Available: https://en.wikipedia.org/wiki/Emotion_classification xiii, 13

[4] A. Kołakowska, A. Landowska, M. Szwoch, W. Szwoch, and M. R. Wróbel, "Modeling emotions for affect-aware applications," *Information Systems Development and Applications*, pp. 55–69, 2015. xiii, 13

[5] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, pp. 37–37, 1996. xiii, 14, 20, 21, 22

[6] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*, 1st ed. Pearson, 2014. xiii, xv, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25

[7] T. M. Mitchell, *Machine Learning*. McGraw-Hill Education, mar 1997. [Online]. Available: http://profsite.um.ac.ir/~monsefi/machine-learning/pdf/Machine-Learning-Tom-Mitchell.pdf xiii, 35, 36, 37

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: http://image.diku.dk/imagecanon/material/cortes_vapnik95.pdf xiii, 37, 38, 39

[9] P. Corr, *The Cambridge handbook of personality psychology*. Cambridge New York: Cambridge University Press, 2009. xv, 5, 6, 8, 12, 13, 14

[10] G. J. Boyle, G. Matthews, and D. H. Saklofske, Eds., *The SAGE handbook of personality theory and assessment*, 1st ed. Los Angeles, CA: SAGE Publications, 2008, oCLC: 244007902. xv, 8, 9

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013. xv, 26

[12] S. Mohammad and S. Kiritchenko, "Using nuances of emotion to identify personality," in *Seventh International AAAI Conference on Weblogs and Social Media*, 2013. 1, 43, 60

[13] R. B. Cattell, "Personality: A systematic theoretical and factual study," 1950. 1, 6

[14] E. Ortiz-Ospina, "The rise of social media." [Online]. Available: https://ourworldindata.org/rise-of-social-media 1

[15] G. W. Allport, "Personality: A psychological interpretation." 1937. 6

[16] D. P. McAdams and J. L. Pals, "A new big five: fundamental principles for an integrative science of personality." *American psychologist*, vol. 61, no. 3, p. 204, 2006. 6

[17] G. W. Allport, "What is a trait of personality?" *The Journal of Abnormal and Social Psychology*, vol. 25, no. 4, p. 368, 1931. 7

[18] R. B. Cattell, "The description of personality: Basic traits resolved into clusters." *The journal of abnormal and social psychology*, vol. 38, no. 4, p. 476, 1943. 8

[19] ——, "The description of personality: Principles and findings in a factor analysis," *The American journal of psychology*, vol. 58, no. 1, pp. 69–90, 1945. 8

[20] R. B. Cattell and H. E. P. Cattell, "Personality structure and the new fifth edition of the 16pf," *Educational and Psychological Measurement*, vol. 55, no. 6, pp. 926–937, 1995. 8

[21] R. L. Piedmont, *The Revised NEO Personality Inventory: Clinical and Research Applications*, 1st ed., ser. The Springer Series in Social/Clinical Psychology. Springer US, 1998. 8

[22] 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. 10

[23] W. T. Norman, "Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings." *The Journal of Abnormal and Social Psychology*, vol. 66, no. 6, p. 574, 1963. 10

[24] J. M. Digman, "Personality structure: Emergence of the five-factor model," *Annual review of psychology*, vol. 41, no. 1, pp. 417–440, 1990. 11

[25] L. Wheeler, *Review of Personality and Social Psychology: Volume 2*, ser. The Review of Personality and Social Psychology. SAGE Publications, 1981. 11

[26] R. Plutchik, *Theories of emotion*. New York: Academic Press, 1980. 12

[27] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion." *Journal of personality and social psychology*, vol. 17, no. 2, p. 124, 1971. 12

[28] C. to Wikimedia, "Feature scaling," Jan 2020. [Online]. Available: http://web.archive.org/web/20200408044337/https://en.wikipedia.org/wiki/Feature_scaling 19, 20

[29] "Cluster analysis," Jun 2020. [Online]. Available: http://web.archive.org/web/20200613145024/https://en.wikipedia.org/wiki/Cluster_analysis 21

[30] "Overfitting: Meaning of overfitting by lexico." [Online]. Available: https://www.lexico.com/definition/overfitting 22

[31] I. Bruha and A. Famili, "Postprocessing in machine learning and data mining," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 2, pp. 110–114, 2000. 23

[32] T. Briscoe, "Introduction to linguistics for natural language processing," *Computer Laboratory, Univ. Cambridge*, vol. 4, 2011. 25, 26

[33] M. T. Pilehvar and J. Camacho-Collados, "Embeddings in natural language processing: Theory and advances in vector representations of meaning," *Synthesis Lectures on Human Language Technologies*, vol. 13, no. 4, pp. 1–175, Nov. 2020. [Online]. Available: https://doi.org/10.2200/s01057ed1v01y202009hlt047 25

[34] "Word embedding," Nov 2020. [Online]. Available: https://en.wikipedia.org/wiki/Word_embedding 26

[35] J. W. Pennebaker and M. E. Francis, "Cognitive, emotional, and language processes in disclosure," *Cognition & Emotion*, vol. 10, no. 6, pp. 601–626, 1996. 27, 42, 43

[36] "What is wordnet?" [Online]. Available: https://wordnet.princeton.edu/ 30

[37] C. Fellbaum, *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. A Bradford Book, may 1998. [Online]. Available: https://www.xarg.org/ref/a/026206197X/ 30

[38] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word–emotion association lexicon," *Computational Intelligence*, vol. 29, no. 3, pp. 436–465, 2013. 30, 31

[39] M. Wilson, "Mrc psycholinguistic database: Machine-usable dictionary, version 2.00," *Behavior research methods, instruments, & computers*, vol. 20, no. 1, pp. 6–10, 1988. 31

[40] S. Poria, A. Gelbukh, A. Hussain, N. Howard, D. Das, and S. Bandyopadhyay, "Enhanced senticnet with affective labels for concept-based opinion mining," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 31–38, 2013. 31

[41] [Online]. Available: https://sentic.net/ 32

[42] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 33

[43] "Natural language toolkit." [Online]. Available: https://www.nltk.org/ 33

[44] M. F. Porter, "An algorithm for suffix stripping," *Program*, 1980. 34

[45] [Online]. Available: https://www.nltk.org/howto/wordnet.html 34

[46] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948. [Online]. Available: https://pure.mpg.de/rest/items/item_2383162/component/file_2456978/content 37

[47] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282. [Online]. Available: https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf 39

[48] [Online]. Available: https://scikit-learn.org/stable/index.html 41

[49] 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. 43, 60

[50] 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. 44, 60, 64

[51] E. P. Tighe, J. C. Ureta, B. A. L. Pollo, C. K. Cheng, and R. de Dios Bulos, “Personality trait classification of essays with the application of feature reduction.” in *SAAIP@IJCAI*, 2016, pp. 22–28. 44

Appendix A

Demonstrations

A.1 Optimal hyperplane's distance calculation

Consider the inequality

$$y_i(w \cdot x_i + b) \geq 1, \quad i = 1, ..., n \ .$$

and the distance formula

$$\rho(w, b) = \underset{\{x:y=1\}}{min} \frac{x \cdot w}{|w|} - \underset{\{x:y=-1\}}{max} \frac{x \cdot w}{|w|}.$$

The inequality can be rewritten as:

$$\begin{aligned} y(w \cdot x + b) &\geq 1 \\ w \cdot x + b &\geq \frac{1}{y} \\ w \cdot x &\geq \frac{1}{y} - b \\ w \cdot x &\geq \frac{1 - by}{y} \end{aligned}$$

The equation for the optimal hyperplane distance maximization can be simplified, using the rewritten inequality, as such:

$$
\begin{aligned}
\rho(w,b) &= \min_{\{x:y=1\}} \frac{x \cdot w}{|w|} - \max_{\{x:y=-1\}} \frac{x \cdot w}{|w|} \\
&= \min_{\{x:y=1\}} \frac{\frac{1-by}{y}}{|w|} - \max_{\{x:y=-1\}} \frac{\frac{1-by}{y}}{|w|} \\
&= \frac{\frac{1-b\times 1}{1}}{|w|} - \frac{\frac{1-b\times(-1)}{-1}}{|w|} \\
&= \frac{1-b}{|w|} - \frac{-1-b}{|w|} \\
&= \frac{1-b-(-1-b)}{|w|} \\
&= \frac{1-b+1+b}{|w|} \\
&= \frac{2}{|w|}
\end{aligned}
$$

Appendix B

Data samples

B.1 Stream-of-Consciousness Essay samples

B.1.1 Sample 1

Well, right now I just woke up from a mid-day nap. It's sort of weird, but ever since I moved to Texas, I have had problems concentrating on things. I remember starting my homework in 10th grade as soon as the clock struck 4 and not stopping until it was done. Of course it was easier, but I still did it. But when I moved here, the homework got a little more challenging and there was a lot more busy work, and so I decided not to spend hours doing it, and just getting by. But the thing was that I always paid attention in class and just plain out knew the stuff, and now that I look back, if I had really worked hard and stayed on track the last two years without getting lazy, I would have been a genius, but hey, that's all good. It's too late to correct the past, but I don't really know how to stay focused n the future. The one thing I know is that when people say that b/c they live on campus they can't concentrate, it's b. s. For me it would be easier there, but alas, I'm living at home under the watchful eye of my parents and a little nagging sister that just nags and nags and nags. You get my point. Another thing is, is that it's just a hassle to have to go all the way back to school to just to go to library to study. I need to move out, but I don't know how to tell them. Don't get me wrong, I see where they're coming from and why they don't want me to move out, but I need to get away and be on my own. They've sheltered me so much and I don't have a worry in the world. The only thing that they ask me to do is keep my room clean and help out with the business once in a while, but I can't even do that. But I need to. But I got enough money from UT to live at a dorm or apartment next semester and I think I'll take advantage of that. But off that topic now, I went to sixth street last night and had a blast. I haven't been there in so long. Now I know why I love Austin so much. When I lived in VA, I used to go up to DC all the time and had a blast, but here, there are so many students running around at night. I just want to have some fun and I know that I am responsible enough to be able to have fun, but keep my priorities straight. Living at home, I can't go out at all without them asking where? with who? why? when are you coming back? and all those questions. I just wish I could be treated like a responsible person for once, but my sister screwed that up for me. She went crazy the second she moved into college and messed up her whole college career by partying too much. And that's the ultimate reason that they don't want me to go and have fun. But I'm

not little anymore, and they need to let me go and explore the world, but I'm Indian; with Indian culture, with Indian values. They go against ""having fun. "" I mean in the sense of meeting people or going out with people or partying or just plain having fun. My school is difficult already, but somehow I think that having more freedom will put more pressure on me to do better in school b/c that's what my parents and ultimately I expect of myself. Well it's been fun writing, I don't know if you go anything out of this writing, but it helped me get some of my thoughts into order. So I hope you had fun reading it and good luck TA's.

B.1.2 Sample 2

Psychologists. Always trying to understand how the mind works, and how it doesn't work in some cases. Can such things be understood, or are we merely deluding ourselves that knowledge of any kind can be attained? I guess I've always found psychology to be a very pretentious field. though an interesting one. We all want to control our lives, and anticipating the actions and desires of others helps us maintain that facade of control. Perhaps I'm getting into a more philosophical realm at the moment, but that is where my thoughts take me. Is free will merely an illusion? I've thought about this a lot. Unfortunately there are no definitive answers to this or other questions. Is there a god? I've never heard a logically sound argument for the existence of a god. I allow for the possibility of a deity, but it certainly wouldn't be the Christian God. I think ultimately that I have to agree with the existentialists. There is no proof for or against the existence of a god, so we should stop wasting time speculating and just deal with this life. Few people can deal with that. Our fear of death makes us create religions, so that we can pretend there is some semblance of life after our earthly bodies die. These are not new thoughts, I'm just thinking on demand; my mind moves most easily to the pathways it knows, and I present some of the more coherent ideas here. Is someone actually reading this? Do you understand that I am human? I am not an object. I am in a body, but I am not the body alone. I am a mind, vast and complex. I am. Do you feel superior because you can analyze minds? I ask you this, so that you can ask yourself. Do you enjoy treating people as objects? Do you even admit in your conscious mind that you treat people as objects? Perhaps not. It's possible that I'm being slightly unfair to you and your profession. Still, it is good to raise questions. We are all just a bit too complacent and easily controlled. I see the need for religion, but I think many of us are above that. I don't need to buy my morality from someone else. What moralists and philosophers do I respect? Plato, for his logic. Kierkegaard, except the theism. Kant, for his explications of metaphysics and epistemology. Nietzsche, except at the end of his days. LaVey, except for his dependence on rituals and his arrogance. Psychologists and behavioral scientists? I stay away from most. At some point I'll get back into it, but I was just too turned off by Freud and his pretentious assumptions. Other reading? Fiction, lots of it. I would name a few dozen authors but then why subject myself to the judgments of someone I can't even see. Music. I find music to be very important. You

can't get by without music. And you can't just listen indiscriminately either. I think a real understanding of notes, rhythms, chords, and instrumentation is required before one can say anything about any kind of music. Do you understand music? How are you reacting to my questions? You must be used to asking the questions instead of having someone else ask them. Are you getting anything out of this? Is this more interesting than most responses to the same assignment, or do you even care? Are you turning to a colleague and saying ""hey, this kid was actually making a futile attempt to understand my motives. "" Fun with role reversals! I considered producing a surreal and rambling narrative for this assignment, but then you might have taken that a bit too seriously (""bob, we got another wacko here""). Ah well, time passes and other pursuits await. Goodbye for now.

B.1.3 Sample 3

I just came back from the Texas Crew Meeting. I sort of want to try out but in the same time very scared. At the meeting, they kept on saying there'll be a lot of hard work and pain. First of all, I don't think I'm in that great of a shape to do such extreme rowing. The crew memebers said it doesn't matter because they will train you from nothing to something. Second of all, I'm a freshman and doing Pre-Pharmacy. I want a good GPA to start off of. I don't know how I'm going to adjust to waking 5 o'clock in the morning. I guess one of the reason I wanted to join is because to challenge myself and make some new friends. More than 100 people from my graduating class comes to UT now. Before school ended, I got very tired of many of the people. I felt they were very fake. I came to UT hoping to make some friends. I still wanted to keep my old friends though. I truly love them. They mean so much to me. What I'm trying to say is that it's hard to find friendships like that. Everyone on my floor seems nice. They smile at me and everything but it seems like they are always in a rush. Everyone has their own things to do and no time to hang out. Anyways, I'm watching American Idol right now and they just announced that Kelli as the American Idol. I never thought the show would be such a hit. WB used to have this show called Pop Star. Not that many people watched it so I wasn't expecting a great rating for this show. The time is almost up so I guess that is it for today. It was very nice writing.