Universidade do Minho
Escola de Engenharia

Inteligência Ambiente: Tecnologias e Aplicações

Trabalho Prático

Mestrado Integrado em Engenharia Informática

1º Semestre
2017-2018
Grupo 8

A75655 - Daniel Camelo Rodrigues
A74219 - Hugo Alves Carvalho
A74702 - José Manuel Gonçalves Leitão da Cunha
A74260 - Luís Miguel da Cunha Lima

13 de Novembro de 2017
Braga

Resumo

Este documento relata o trabalho prático desenvolvido no âmbito da unidade curricular de **Inteligência Ambiente: Tecnologias e Aplicações**, encontrando-se dividido em duas partes: uma primeira associada à análise comportamental e uma segunda associada à análise sentimental.

Conteúdo

1 Introdução

É comum a ideia de que, quando a mente humana entra em ação, o pensamento se forma em primeiro lugar. Mas, numa camada mais profunda do que aquela que em que se forma o pensamento, surge o sentimento, responsável por gerar um determinado pensamento.

"As pessoas pensam porque sentem."

A força criativa não é acionada diretamente pelo pensamento uma vez que esta é decorrente de um sentimento. Assim sendo, os sentimentos revelam um papel de extrema importância pois estão na origem de todos os pensamentos e ações.

Nos últimos anos, com o objetivo de manter as suas atividades atualizadas no mercado, várias marcas e empresas se preocupam em utilizar fatores influenciáveis no ser humano que permitam um melhor rendimento dos serviços prestados. Neste contexto, torna-se importante poder avaliar a satisfação do cliente em relação a esses serviços. É importante medir a satisfação do cliente, ou seja, a diferença entre as suas expectativas e o desempenho do serviço realmente entregue. No entanto, devido ao alto volume de dados a serem analisados, é praticamente impraticável avaliar tudo manualmente.

Nesse contexto, surge a Análise Sentimental, que é o conjunto de técnicas da computação utilizadas para extrair, classificar, compreender e avaliar os sentimentos e opiniões expressas pelos utilizadores em fontes textuais. Pode ser usado, por exemplo, para compreender as opiniões dos eleitores sobre eventos políticos ou as opiniões dos consumidores sobre os produtos de uma empresa. A opinião ou os sentimentos de uma pessoa são, em sua maior parte, subjetivos e não factos. O que significa que analisar com precisão a opinião ou o humor de um indivíduo de um texto pode ser extremamente difícil. Com *Sentiment Analysis*, a partir de um ponto de vista da análise de texto, estamos essencialmente a procurar obter uma compreensão da atitude de um indivíduo em relação a um tópico num texto e a sua polaridade, seja esta positiva, negativa ou neutra.

Embora o sentimento e a análise de comportamento sejam muito úteis para desenhar o comportamento dos utilizadores em determinados produtos ou eventos, é necessário monitorizar e prever qualquer incerteza, identificando estratégias de como abordar e lidar com estas situações.

2 Parte 1: Análise Comportamental

2.1 Descrição do Problema

O primeiro trabalho prático da unidade curricular de *Inteligência Ambiente: Tecnologias e Aplicações* tem como objetivo analisar um ficheiro log com padrões de comportamento de escrita de um indivíduo, através da sua interação e uso do teclado.

Deste modo, o grupo elaborou um projeto na linguagem de programação **C#** onde foi desenvolvido um conjunto de métricas que permitem analisar informações que posteriormente serão úteis para interpretar padrões associados a um determinado utilizador. Ao longo desta secção serão explicadas todas as métricas de análise comportamental implementadas na aplicação, os algoritmos utilizados e a análise dos resultados obtidos.

2.2 Métricas

2.2.1 Análise de Eventos *Keystroke*

O primeiro passo para conseguirmos implementar um conjunto de métricas correspondentes a Eventos *Keystroke* passou por analisar como estavam estruturados os ficheiros log e como poderiam ser recolhidas as informações necessárias.

/ [Timestamp]:[Event]:[Character]: */*

692547859:KeyPressed:O

692548015:KeyPressed:L

692548125:KeyPressed:A

692548187:KeyPressed:Space

Em seguida, os dados foram guardados em memória numa lista, e implementadas as seguintes métricas relativas a eventos singulares com o teclado:

- Número total de caracteres utilizados
  - corresponde ao número total de linhas registadas no ficheiro e guardadas na lista.
- Número total de vezes que o caracter *backspace* foi utilizado e sua respetiva percentagem de uso
  - consiste em contar quantas vezes o caracter "anterior"foi registado e fazer a sua relação com o número total de caracteres. Esta métrica permite identificar o nível de incorreção de um utilizador durante a sua escrita.
- Top 10 de caracteres mais utilizados, número total de vezes que cada um foi utilizado e sua percentagem de uso.
  - percorrendo a lista de caracteres e utilizando um *dictionary*, é possível contar quantas vezes cada caracter é usado. Como é importante verificar quais os caracteres mais utilizados, a lista é ordenada e são identificados os dez mais relevantes, bem como a sua relação com o número total de caracteres utilizados.

2.2.2 Análise de Eventos *Digraph*

Após desenvolvidas as métricas associadas a eventos *keystroke*, torna-se agora importante calcular e interpretar a latência de todos os eventos registados anteriormente.

Um fator que influencia a variação de intervalos de tempo entre dois eventos é a posição que cada caracter ocupa no teclado. Assim, pretendendo obter uma informação mais precisa e concreta, podemos dividir o teclado em três grupos: *Left-button, Right-button e Space-button.*

Figura 1: Layout do teclado

Percorrendo a lista com cada registo do ficheiro log, foi então criada numa nova lista que em cada registo contém a latência (diferença entre dois valores de *timestamp*), os dois caracteres utilizados e o seu *keygroup pair*.

[ΔTime]:[1stKey]:[2ndKey]:[KeyGroup Pair]

Já com os novos dados guardados em memória, foram então implementadas as novas métricas de análise:

- Média do tempo de escrita
  - percorrendo todos os registos da nova lista (1º campo), obtemos o valor médio de escrita de um utilizador.
- Desvio padrão do tempo de escrita
  - permite verificar o nível de variação no tempo de escrita entre todos os eventos registados.
- Análise baseada em *KeyGroup Pairs*: número de vezes que cada um dos pares foi utilizado, percentagem de uso, média de escrita e desvio padrão.
- Top10 do uso de KeyGroup Trios: número de vezes que cada um foi utilizado e a sua percentagem de uso.
  - para além de serem analisados os eventos associados a conjuntos de pares, o grupo achou importante verificar também quais os eventos em trio mais utilizados. Seguindo o mesmo raciocínio de identificação dos pares, foi então possível assinalar os padrões mais frequentes do utilizador.

2.2.3 Análise de Eventos de Palavras

Pretende-se agora que seja possível realizar um conjunto de métricas sobre uma análise de eventos de palavras. Para tal, é necessário a identificação correta das palavras a analisar.

Numa primeira abordagem, é necessário categorizar os carateres que são recebidos no log que contêm informação sobre o tempo do carater pressionado. Os carateres podem ser de dois tipos: delimitadores ou não-delimitadores. Os carateres delimitadores são aqueles que não assinalam um carater de uma palavra, sendo os não-delimitadores precisamente o contrário (exemplo de delimitadores:" Oemcomma","

Oemperiod"," Tab"). A análise é realizada a partir da lista logs e para a sua realização, necessitamos das seguintes listas de strings:

```
 /* Lista para os carateres */
List<string> characters = null;
/* Lista para os tempos */
List<string> time = null;
/* Lista para a flag do backspace */
List<string> backspaces = null;
/* Flag backspace(utilizado-1 |não utilizado-0) */
int backspace = 0;
/* Lista que regista tudo do log */
List<string> logger = new List<string>();
```

Iterando o ciclo referente ao log onde se encontram as informações dos tempos dos carateres pressionados, começamos por realizar um *split* para obtermos as componentes da linha separadamente e realizarmos operações sobre as mesmas(*timestamp,character*).

Para cada iteração do ciclo, procuramos saber se nos encontramos ou não no fim do processamento. Por um lado, caso nos encontremos no fim do ciclo, apenas adicionamos um carater terminador às listas correspondentes na ocorrência de este pertencer ao tipo delimitador. Se for do tipo não delimitador, adicionamos o carater identificado às listas que dizem respeito. Por outro lado, investiga-se o carater que se segue. Para este caso, existem as seguintes condições:

- O que acontece se o carater atual for um delimitador?
  - Adiciona-se o carater ao logger.
- O que acontece se o carater atual for um "Anterior"?
  - Neste caso, significa que iremos apagar das listas o último carater. Se este carater corresponder a um carater separador na lista dos *characters, timestamp ou backspaces* apagamos duas vezes pois pretendemos apenas apagar os carateres não separadores.
- O que acontece se o carater atual for um delimitador e o seguinte for um carater não delimitador?
  - Identificamos este caso como o caso onde iremos estabelecer o final da palavra, adicionando os carateres separadores às listas mencionadas.
- O que acontece se o carater atual corresponder a um carater não delimitador?
  - Adicionamos às listas dos *characters* o carater e o seu tempo do à lista dos *timestamps*.

No final do ciclo, obteremos as três listas com a informação pertencente ao log separadas por ";". Antes de a função **Words()** (responsável por este algoritmo de criação e identificação de palavras) terminar, é ainda realizada uma passagem desta informação organizada no formato pretendido através de uma lista de strings *listWords*.

Com base neste procedimento de identificação de palavras, passou-se então à implementação das métricas de análise deste tipo de eventos:

- Número total de palavras escritas
- Número total de palavras em que o backspace foi utilizado
  - permite assim verificar a correção de escrita do utilizador, relacionando o número de palavras que tiveram erros durante o sue processamento com o número total de palavras escritas.
- Média de tempo de escrita

  - com a obtenção deste valor, é possível identificar o nível de rapidez com que um utilizador escreve no teclado.

- Desvio padrão do tempo de escrita
- Análise de Palavras por tamanho: total de palavras por tamanho, percentagem de uso de backspace, média e desvio padrão do tempo de escrita.
  - o valor do desvio padrão tende a ser elevado pois um utilizador escreve várias palavras com tamanho distinto e consequentemente com tempo de escrita muito diferente. Deste modo, torna-se importante analisar também o padrão de escrita do indivíduo perante palavras do mesmo tamanho.
- Top 10 de palavras mais utilizadas: número de vezes que cada palavra foi escrita e sua respetiva percentagem de utilização

2.3 Interface

Com o objetivo de apresentar ao utilizador uma aplicação simples e intuitiva, foi desenvolvida uma interface que permite a execução de todas as métricas de análise mencionadas anteriormente. Em primeiro lugar, deve ser carregado o ficheiro log:

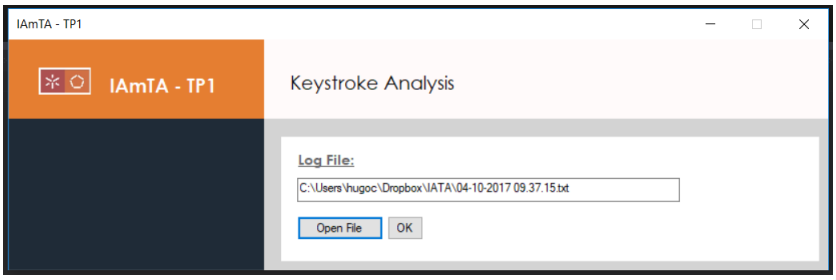

Figura 2: Janela inicial para carregamento de ficheiro log

Após carregado o ficheiro, a aplicação executa as funções de análise e os resultados são apresentados. A interface encontra-se agora dividida em três divisões: Eventos *Keystroke*, Eventos *Digraph*, Eventos de Palavras.

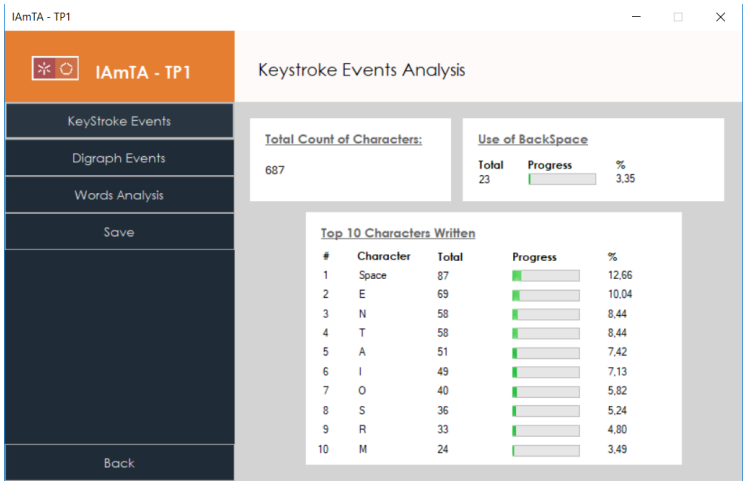

Figura 3: Janela de resultados para eventos *keystroke*

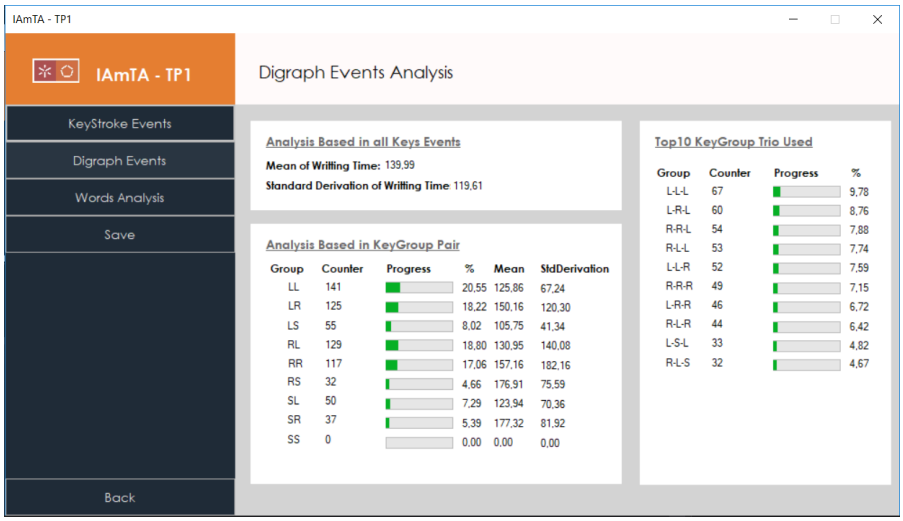

Figura 4: Janela de resultados para eventos *digraph*

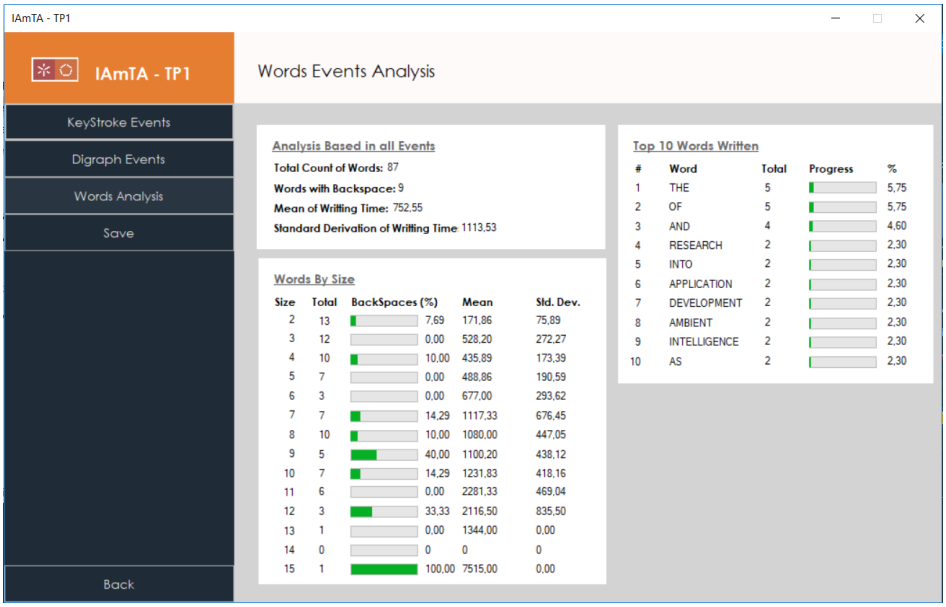

Figura 5: Janela de resultados para eventos de palavras

Para além de serem apresentados os resultados para todas as métricas desenvolvidas, a interface oferece também a possibilidade do utilizador gravar toda a análise numa base de dados, associando o seu nome de utilizador e data.

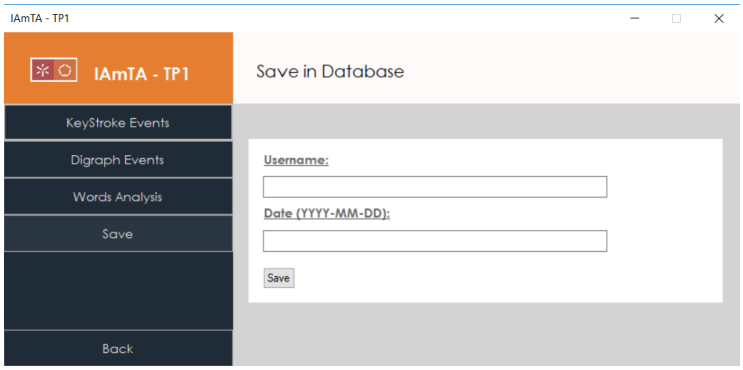

Figura 6: Janela de resultados para eventos de palavras

2.4 Análise de Resultados

Após a apresentação dos resultados das métricas desenvolvidas, prosseguiremos à sua análise para consolidar o desempenho do sistema desenvolvido.

Analisando os resultados obtidos para os eventos *keystroke* é essencial interpretar os resultados das percentagens apresentadas. Os valores associados ao uso caracter "backspace"são importantes pois permitem verificar o nível de incorreção do utilizador numa determinada sessão.

Na janela dos eventos *digraph* é fundamental analisar os valores do tempo médio e desvio padrão do tempo de escrita pois permitem verificar a variância de escrita consoante os diferentes padrões utilizados no teclado. Na análise da utilização de cada grupo (pares ou trios) facilmente se repara que os grupos onde o caracter "space"está inserido, são os que apresentam menor utilização, tal como seria de esperar. Por outro lado, todos os restantes grupos de equilibram entre si na percentagem de uso.

Explorando os resultados associados aos eventos de palavras, é possível observar que o valor do desvio padrão é muito elevado pois o utilizador escreve palavras com tamanhos distintos, logo o tempo de escrita sofre uma grande variação. Analisando os mesmos valores para cada tamanho de palavra, naturalmente se verifica que o tempo de escrita cresce praticamente de forma linear à medida que o tamanho da palavra aumenta. Por outro lado, é de notar o mesmo crescimento face ao uso do caracter "backspace".

Todos os valores apresentados são de extrema importância para a identificação de padrões associados a um determinado indivíduo. Com todos estes dados guardados numa base de dados, é possível mais tarde identificar sintomas de fadiga ou stress sempre que os valores de um utilizador variarem dos seus resultados médios.

3 Parte 2: Análise Sentimental

3.1 Descrição do Problema

"As opiniões e os seus conceitos relacionados, tais como sentimentos, avaliações, atitudes e emoções, são temas de análise de sentimentos e descoberta de conhecimento. O rápido crescimento desta área coincide com as necessidades dos social media na Web, relativamente à analise automática de comentários, discussões em fórum, blogs, microblogs e redes sociais. Isto sucede visto que pela primeira vez na história do ser humano, existe o acesso a um enorme volume de dados de opiniões registadas em formato digital."

O segundo trabalho prático da unidade curricular de *Inteligência Ambiente: Tecnologias e Aplicações*, tem como objetivo a análise sentimental do(s) texto(s) de um determinado utilizador através de métricas específicas. Pretende-se que se obtenha um conhecimento geral do sentimento transmitido sobre as oito emoções a serem processadas: *Raiva, Antecipação, Desgosto, Medo, Alegria, Tristeza, Surpresa e Confiança.*

O presente sistema solucionado procederá, de forma sucinta, à apresentação das funcionalidades que permitam a identificação de palavras mais relevantes no texto (palavras-chave). Para tal, a implementação realizar-se-á por intermédio de técnicas TF-IDF (*Term Frequency-Inverse Document Frequency*) (Salton and Buckley, 1988). Com base no conjunto obtido será elaborado uma análise sentimental do mesmo identificando as relações / dependências entre emoções e padrões de escrita de um utilizador.

3.2 Métricas

3.2.1 *Stop Words*

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

Figura 7: Exemplo de uma lista de StopWords

Em processos de análise textual, é frequente a existência de palavras sem valor no objetivo da avaliação a realizar, isto é, que não constituem interesse face à necessidade do utilizador e de tal modo são excluídas. Essas palavras são designadas de StopWords. A estratégia geral para a sua identificação, consiste no cálculo da frequência dos termos apresentados seguida de uma ordenação. Posteriormente, as palavras são filtradas estando finalizadas para uma subsequente operação sobre as métricas pretendidas. Um exemplo de uma lista de StopWords encontra-se representada na Figura 2.

3.2.2 Identificação das *Stop Words*

As *stopwords* são identificadas no(s) documento(s), por intermédio de técnicas TF-IDF (*Term Frequency-Inverse Document Frequency*). Esta constitui uma medida estatística que tem o intuito de indicar a importância de uma palavra de um documento em relação a uma coleção de documentos.

Em primeiro lugar, para cada documento selecionado através da interface gráfica, executa-se uma tokenização. Este pré-processamento constitui o passo mais importante no algoritmo. A informação no mundo atual constata-se em 90% das vezes de incompleta, incorreta e inconsistente. Para tal, torna-se fundamental a aplicação deste procedimento.

```
List<string> tokens = new List<string>();
Tokenizer tokenizer = new Tokenizer();
tokens = tokenizer.Tokenize(text);
```

Figura 8: Pré processamento de tokenização

De modo a obter o IDF de cada palavra, desenvolve-se um dicionário cuja chave representa o nome da palavra e o valor traduz o número de vezes que esta aparece nos documentos selecionados pelo utilizador. No final, obteremos a frequência inversa do documento correspondente á palavra em questão. De modo análogo para o cálculo da métrica TF, concebe-se um dicionário em que a chave retrata o nome da palavra e o valor exprime o número de vezes que a mesma aparece no documento em questão. Para concluir o algoritmo, multiplicando os valores de ambos dicionários adquirimos a métrica TF-IDF para as palavras em questão, possibilitando o conhecimento dos termos a remover.

3.2.3 Identificação do Sentimento

A funcionalidade principal do sistema consiste, na análise do sentimento das palavras. Esta análise baseia-se no modelo de 8 emoções básicas proposto por Plutchik (1980) aplicando o léxico EmoLex 1, no qual relaciona as palavras contidas no texto com as emoções associadas (Mohammad and Turney, 2013). Utilizaremos este modelo apenas para a língua inglesa uma vez que, representa a língua universal.

Com destino a utilizar e manipular os dados em C, exportamos a informação em Excel para um ficheiro *csv*. *LumenWorksCsvReader* representa a biblioteca usufruída para ler os dados em *csv* no programa desenvolvido. Produzimos o dicionário *keywordDictionary*, com a finalidade de para cada palavra, guardar a lista de número que dizem respeito ao seu sentimento.

```
static void StoreCSVInfo(string s)
{
    keywordDictionary = new Dictionary<string, List<int>>();
    using (CsvReader csv = new CsvReader(new StreamReader(s), true))
    {

        int fieldCount = csv.FieldCount;
        string[] headers = csv.GetFieldHeaders();

        while (csv.ReadNextRecord())
        {
            List<int> numbers = new List<int>();
            string[]part = csv[0].Split(';');

            for (int i = 1; i < part.Count(); i++)
            {

                if (part[i] == "0" || part[i] == "1")
                numbers.Add(Convert.ToInt32(part[i]));
            }
            keywordDictionary.Add(part[0], numbers);
        }
    }
}
```

Figura 9: Função responsável por guardar os dados do ficheiro *csv* no dicionário *keywordDictionary*.

Por último, para cada palavra no(s) documento(s), calculamos o seu sentimento caso estas existam no keywordDictionary. Se existirem, para cada número com o valor 1 na lista de associada a cada palavra no dicionário, incrementamos o contador do sentimento respetivo.

3.3 Interface

O sistema desenvolvido apresenta uma interface gráfica, que permite a execução das métricas relativas á análise dos sentimentos. O utilizador dispõe da decisão de optar por um texto previamente produzido, ou compor um novo.

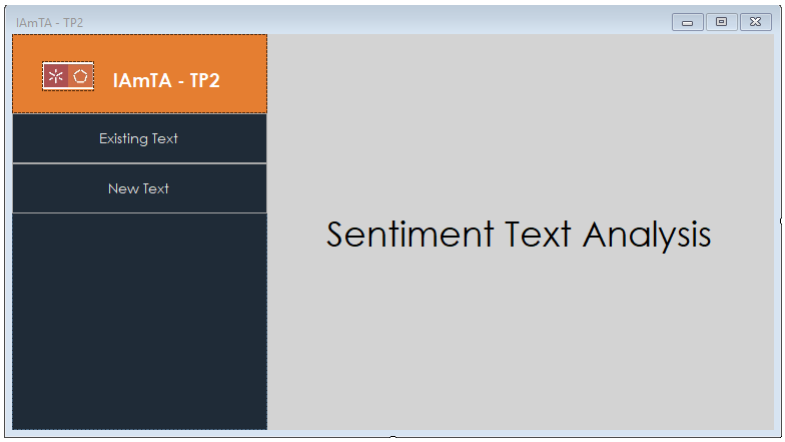

Figura 10: Ecrã inicial do sistema desenvolvido

Caso o utilizador pretenda digitar um novo texto, será apresentado um espaço para concretizar a sua ação. Contrariamente, é lhe solicitado que abra um ficheiro de texto previamente composto.

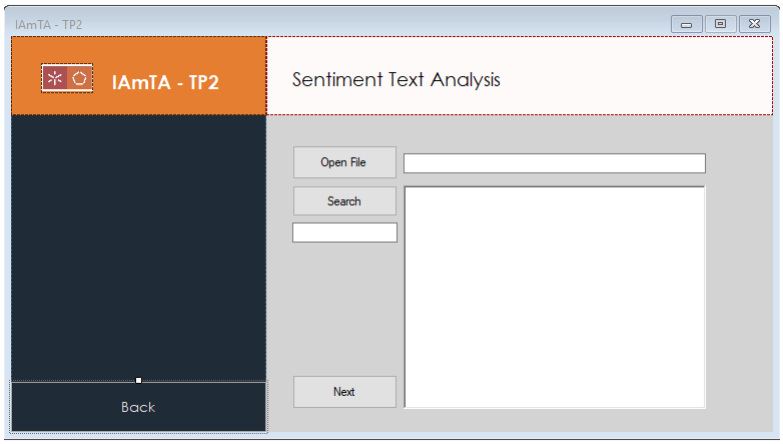

Figura 11: Ecrã após a decisão de escolher um texto previamente composto

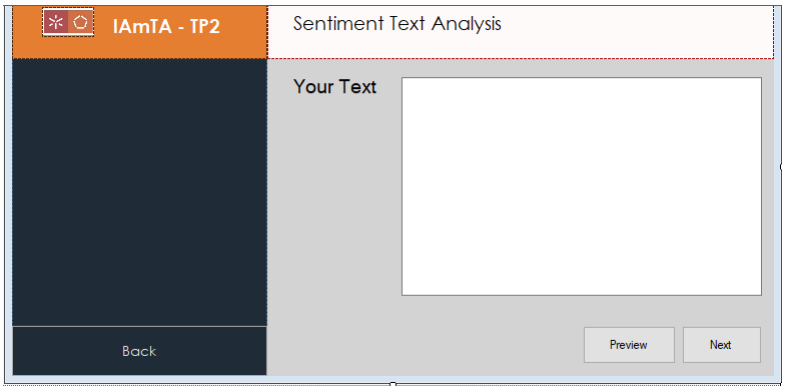

Figura 12: Ecrã após decisão de compor um novo texto

De seguida, são apresentadas as 10 palavras com maior percentagem de IDF nos documentos. Além disso, o utilizador dispõe da opção de escolha das *stopwords* a remover. Caso pretenda, deverá assinalar as palavras a excluir e pressionar o botão *"Get Items/Next"*. O utilizador também poderá escolher a avaliação sentimental por *stemming* ou bi-gram.

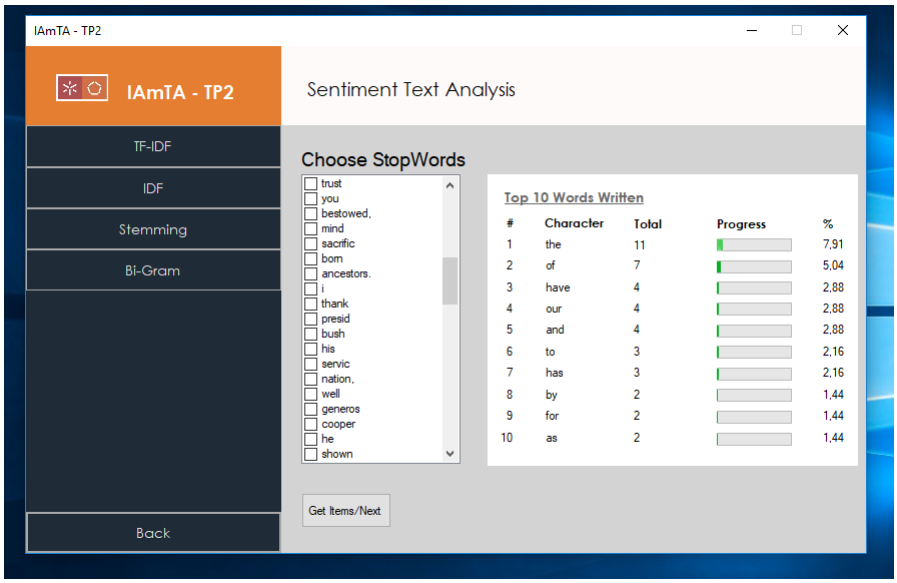

Figura 13: Ecrã da percentagem IDF e seleção das *stopwords* (o exemplo mostrado é com *stemming*).

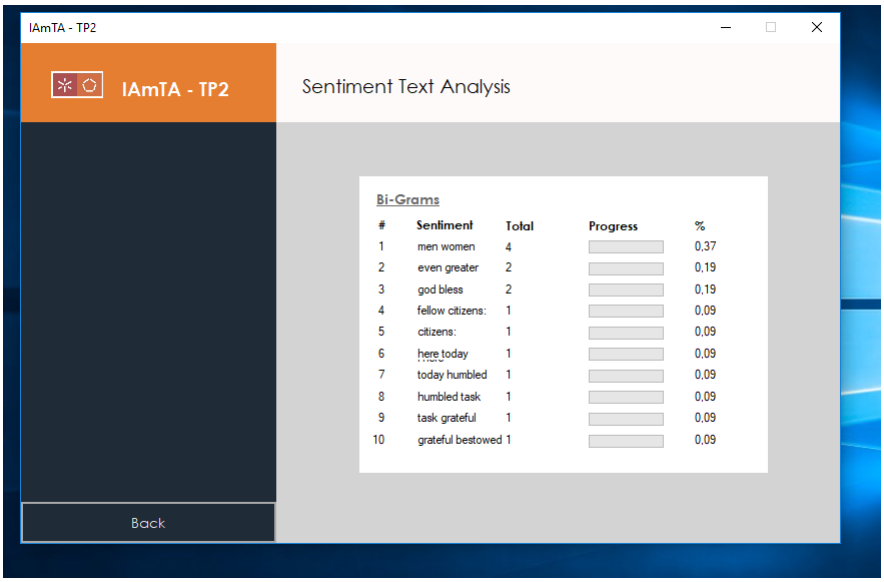

Figura 14: Ecrã da análise em bi-gram.

Após a decisão do utilizador, se pressionar o botão "TF-IDF", a interface dispõe das 10 palavras com maior percentagem TF-IDF.

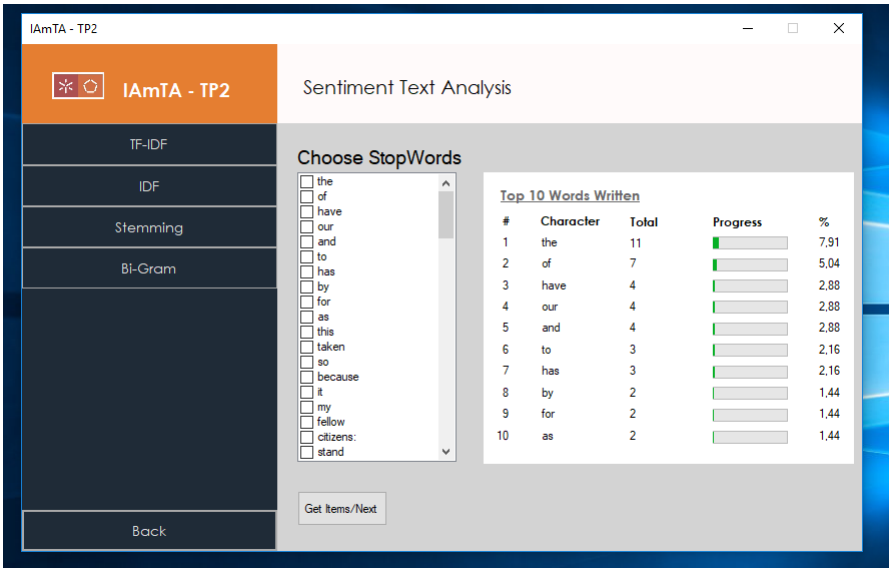

Figura 15: Ecrã com a métrica TF-IDF.

Por fim, apresenta-se a análise sentimental do texto das 8 emoções: *Raiva, Antecipação, Desgosto, Medo, Alegria, Tristeza, Surpresa e Confiança.* Esta análise é concluída com a percentagem de cada sentimento.

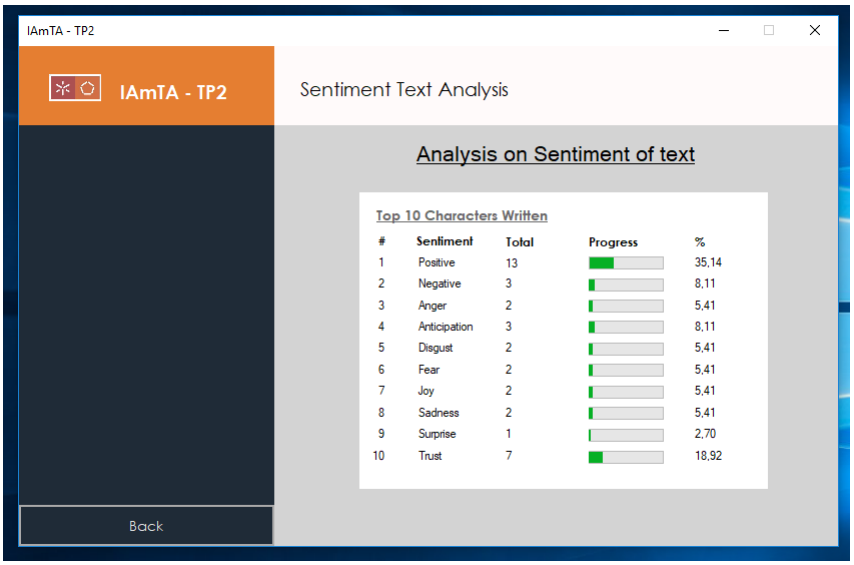

Figura 16: Ecrã contendo a informação da análise dos sentimentos.

3.4 Análise de Resultados

Após a apresentação dos resultados das métricas desenvolvidas, prosseguiremos à sua análise para consolidar o desempenho do sistema desenvolvido.

Na Figura 15, apresenta-se a informação obtida utilizando a métrica TF-IDF. Ao analisar os dados, constata-se que as palavras com maior percentagem podem ser classificadas como stopwords uma vez que, estas palavras não evidenciam algum tipo de sentimento. Estes termos são removidos na hipótese de o utilizador as escolher. Obteve-se vocábulos como "the"," of", "and" estando de encontro aos resultados que se expectava. Estas palavras são as mais repetidas no(s) documento(s) e deste modo, exibem uma percentagem elevada de TF-IDF.

A Figura 16 evidencia a análise dos sentimentos transmitidos dos documentos escolhidos pelo utilizador. Denota-se que os sentimentos com maior percentagem são de *Positive e Trust*. O texto revela 35.14% de sentimento positivo e 18.92% de confiança. Estes resultados, são capazes de inferir algumas informações sobre o caráter ou personalidade do autor dos documentos. Existindo um padrão positivo e de confiança, é possível concluir que o compositor é uma pessoa otimista e com um grau moderado de credibilidade sobre os assuntos discutidos nos diversos manuscritos por si escritos. Será possível então concluir esta afirmação baseado nestes documentos? Na verdade, não existe 100% confiança que a pessoa responsável por estes textos disponha desta personalidade, uma vez que é possível que haja outras publicações em que o sentimento seja diferente do positivo e de confiança. No entanto, estes resultados constituem a importância da análise sentimental. Somos capazes de compreender o caráter da pessoa sem a conhecer, só mesmo tomando conhecimento dos seus documentos.

4 Conclusão

O primeiro trabalho realizado foi de acordo com o grupo, um trabalho conseguido e fundamentado. Achamos que a nossa implementação das métricas de ambas as partes relativas á analise comportamental e sentimental, são adequadas e atingem os objetivos pretendidos pelo enunciado.

Em relação á primeira parte, consideramos que a implementação da análise de eventos keystroke e digraph foram de relativa facilidade estando na nossa opinião os objetivos propostos dado como cumpridos. No entanto, poderíamos ter implementado novas métricas constituindo novas ideias para esta análise. Contudo, achamos que as que foram calculadas são as mais importantes e relevantes para o problema em questão. No que diz respeito á análise das palavras, compreendemos que o nosso algoritmo não seja o melhor para a sua identificação, mas cumpre o objetivo pretendido, calculando-as corretamente e portanto, realizamos uma apreciação positiva desta fase no geral.

Por outro lado, em relação á segunda parte, avaliamos esta fase num tom positivo, não identificando qualquer tipo de contratempo ou falha. Consideramos que a análise sentimental foi conseguida obtendo resultados satisfatórios.

De um modo geral, consideramos que os objetivos deste primeiro trabalho foram cumpridos com sucesso, estando o grupo preparado para o segundo trabalho, esperando ter cumprido o que lhes foi pedido até a data sem falhas relevantes.

5 Referências Bibliográficas

1. Damashek, M. (1995). Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267(5199):843.

2. Gunetti, D. and Picardi, C. (2005). Keystroke analysis of free text. *ACMTransactions on Information and SystemSecurity (TISSEC)*.

3. Lo, R. T.-W., He, B., and Ounis, I. (2005). Automatically building a stopword list for an information retrieval system. In *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information RetrievalWorkshop (DIR), volume 5, pages 17–24*

4. Mohammad, S.M. and Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon.

5. Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*.

6. Plutchik, R. (1980). *Emotion: Theory, Research, and Experience. Ed. by Robert Plutchik, Henry Kellerman.* Number v. 0. Acad. Press.

7. Porter,M. F. (2001). Snowball: A language for stemming algorithms.

8. Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing management*.