Departamento de Ciências e Tecnologias da Informação

Text Mining from Curricula

Tiago João Aires Soares

Dissertação submetida como requisito parcial para obtenção do grau de
Mestre em Informática e Gestão

Orientador(a):
Doutora Ana Maria de Almeida, Professora Auxiliar,
ISCTE-IUL

Coorientador(a):
Doutor Fernando Manuel Marques Batista, Professor Auxiliar,
ISCTE-IUL

Outubro, 2018

Agradecimentos

Ao meu orientador e coorientador, Professora Ana Maria de Almeida e Professor Fernando Manuel Marques Batista, pelo total apoio e opiniões críticas.

À família, e em especial aos meus pais, pelo apoio incondicional, económico e emocional, ao longo de todo este percurso académico.

À minha namorada, pela força e apoio nos momentos mais difíceis.

A todos os meus amigos, que sabem quem são, por todos os conselhos.

A todos os que acima referi dedico este trabalho, e o meu sincero Muito Obrigado.

Resumo

O processo de recrutamento de candidatos é um tema que está presente em todas as empresas nos dias de hoje, sendo esta uma vertente essencial para um bom funcionamento de qualquer empresa que preze pela contratação dos melhores candidatos possíveis. A evolução tecnológica ao longo das décadas tem obrigado a mudanças constantes no processo de recrutamento, visto que, com a utilização cada vez maior da Internet, o crescimento da existência de dados e, por consequência, a informação nela contida, aumenta diariamente, tornando-se impossível, mas desejável, acompanhar toda a informação útil. Por essa razão hoje em dia a maior parte das empresas utiliza a Internet como uma ferramenta necessária para o seu processo de recrutamento e, por isso, são cada vez mais utilizadas técnicas de *Text Mining* (TM) para a contratação de candidatos, agilizando assim o processo de recrutamento, tornando-o mais eficiente, e gastando menos recursos, quando comparado a um processo tradicional. Ao utilizarmos técnicas de TM num processo de recrutamento face a um processo tradicional estamos não só a reduzir o tempo gasto com cada candidato, como também a reduzir os custos inerentes, isto é, podemos obter o melhor candidato possível por um menor custo face ao antigo processo. Imaginando um universo de 1000 candidatos, ao utilizar um processo de recrutamento tradicional, estaríamos a gastar recursos com a leitura de 1000 candidaturas e, possivelmente, igual número de entrevistas, o que não aconteceria com a utilização de um processo com recurso a *Text Mining*, pois não seriam gastos recursos com a leitura inicial dos CV, e apenas seriam lidos os escolhidos como melhores candidatos para uma fase de entrevista. Em suma, este processo de recrutamento está a ser adotado pela maior parte das empresas em todo o mundo pois é um processo com inúmeras vantagens, quer para o recrutador, quer para o candidato. Entre as vantagens incluem-se a redução de custos por candidato, a um maior alcance geográfico dos candidatos, à redução de tempo gasto no processo, a uma maior precisão nos candidatos alvo, entre outros.

Palavras-Chave: Text Mining, Recursos Humanos, Recrutamento eletrónico, Candidatos, Análise de CV.

Abstract

The process of recruiting candidates is not only a theme that is present in every company nowadays, but also an essential aspect for a smooth operation of any company that seeks to hire the best candidates possible. The technological evolution over the decades has forced constant changes in the recruitment process given that, with the increasing Internet usage, the growth of the amount of data and information contained therein increase daily, making it impossible, although desirable, to track all the useful information. For this reason, most companies use Internet as a strong tool for their recruitment process and, therefore, *Text Mining* techniques are being increasingly used to recruit candidates thus streamlining the recruitment process, making it more efficient, and spending less resources, in contrast to a traditional process. By using TM techniques in a recruitment process instead of a traditional process we are not only reducing the time spent with each candidate, but also reducing the inherent costs, in other words, we can get the best possible candidate for a lower cost than the old process. Imagining a range of 1000 candidates, using a traditional recruiting process, we would be spending resources on reading 1000 applications, and maybe doing 1000 interviews, this would not be the case when using a *Text Mining* process, since there would be no expenses with the initial reading of the applications, and only those chosen as the best candidates for an interview phase would be read. In short, this recruitment process is being adopted by most companies around the world as it is a process with numerous advantages for both the recruiter and the candidate. The benefits range from reducing costs per candidate, to a greater geographic reach of candidates, to reducing the time spent in the process, to greater Accuracy in target candidates, among others.

Keywords: Text Mining; Human Resources; E-recruitment; Candidates; CV analysis.

Índice

Índice de Tabelas

Índice de Figuras

Lista de Abreviaturas e Siglas

CV – Curriculum Vitae

HMM – Hidden Markov Model

LDA – Latent Dirichlet Allocation

LSI – Latent Semantic Analysis

NMF – Non-Negative Matrix Factorization

SVM – Support Vector Machine

TM – Text Mining

Capítulo 1. Introdução

Num mundo em constante evolução e cada vez mais automatizado, até nos mais pequenos pormenores do dia a dia, esta dissertação surge numa tentativa de contribuir positivamente para essa evolução, assim sendo a presente dissertação teve por base uma proposta de tema por parte da Siemens Portugal: Text Mining From Curricula. Este tipo de abordagem a problemas de tratamento de texto começa a ser cada vez mais popular devido às suas inúmeras vantagens, podendo ser adaptado aos mais diversos temas e âmbitos, desde problemas simples como a análise de sentimentos presentes numa frase, a complexos como análise de relatórios da indústria biomédica.

Este tema reflete assim um assunto procurado por muitas das empresas nos dias de hoje, uma ferramenta automática de tratamento de CVs que selecione os melhores candidatos. Com uma ferramenta como esta seria possível uma empresa processar a um maior ritmo os CVs de possíveis candidatos, reduzindo assim o tempo de espera de uma resposta, reduzir os custos com um departamento de Recursos Humanos, e toda a sua logística, escolher os melhores candidatos para um, entre outras vantagens.

1.1. Motivação

Sendo Data Mining um tema em grande expansão e aposta a nível global nos dias de hoje devido ao constante avanço tecnológico, a principal motivação para a realização desta dissertação de mestrado neste tema, Text Mining – uma derivação de Data Mining – assenta na contínua curiosidade acerca dessa evolução. Devido aos avanços tecnológicos deste século, cada vez mais irão deixar de ser utilizados métodos tradicionais de realizar tarefas, para se começarem a usar métodos automatizados, tal como Text Mining, de modo a preservar recursos essenciais para as empresas. E daí surge a vontade de querer criar uma ferramenta útil de análise de currículos que possa ser utilizada num ambiente real de uma empresa.

1.2. Objetivos

Pretende-se com esta dissertação fazer uma análise a diversos currículos, concretamente aos candidatos mais promissores a serem contratados para um certo cargo,

de modo a automatizar o processo de recrutamento com recurso a técnicas de TM. Para tal 1000 CVs reais de uma empresa serão analisados na esperança de encontrar informações relevantes que possam levar, ou não, à contratação de um candidato.

Por outras palavras, esta dissertação tem como principal objetivo concluir, de acordo com o seu CV, se um candidato é ou não adequado ao cargo ao qual se está a candidatar, utilizando técnicas de TM.

1.3. Enquadramento e Problema

O problema que este estudo se propõe a resolver passa por extrair informação de vários currículos de candidatos, utilizando técnicas de TM para que estes possam ser avaliados e, posteriormente, aceites ou não para um certo cargo.

Numa primeira fase é necessário que todos os currículos se encontrem no mesmo formato, logo é necessária uma uniformização dos dados. Na maioria dos dados de um currículo a informação tem uma forma semiestruturada, pelo que é necessário colocá-la num formato estruturado.

É também fundamental para este problema categorizar a informação contida nos currículos para, posteriormente, ser extraída em blocos já categorizados, isto é, um currículo é composto por vários blocos de informação sobre o candidato tais como:

- Informação pessoal:
    - Nome
    - Morada
    - Número de telefone
    - Género
    - Data de nascimento
- Educação
    - Escola secundária frequentada
    - Faculdade frequentada
    - Curso frequentado
- Experiência profissional
    - Empresas anteriores
    - Duração do trabalho
    - Cargo que ocupou

- Projetos que esteve inserido

Extrair a informação de maneira a que esta possa ser trabalhada para atingir o objectivo final tem uma preponderância fulcral. Após a categorização e estruturação da informação irão então ser testadas diversas maneiras para possível resolução do problema, categorização automática de um CV de acordo com o cargo ao qual se está a candidatar.

1.4. Abordagem

Apresentado o problema, é necessário abordar as várias fases que o compõe. Como já foi referido no ponto 1.3) um currículo recebido contém vários campos que representam um candidato: informação pessoal, formação académica, experiência profissional e ainda outras informações que o candidato considere relevante anexar ao seu currículo.

Um candidato, ao criar o seu currículo, escolhe o formato que pretender. Logo, em cerca de 1000 candidatos irão existir currículos em diversos formatos. A primeira fase da abordagem a este estudo irá tratar o problema da diversificação de formatos que podem existir, convertendo todos esses formatos para um único, sem qualquer perda de dados, de forma a tornar possível um tratamento mais eficiente.

De seguida é necessário classificar e dividir os vários campos que compõe um CV, isto é, segmentar o currículo por tópicos, para que seja mais fácil o tratamento da informação contida em cada tópico.

Com a informação já segmentada pelos diferentes tópicos irá ser necessário extrair apenas a informação relevante contida em cada um dos blocos, para que esta possa ser utilizada aquando da escolha de um candidato.

1.5. Estrutura do Documento

Este documento está dividido em seis capítulos, sendo que o presente capítulo corresponde à contextualização do tema, contendo a motivação que levou à concretização desta dissertação, aos objetivos da mesma, ao enquadramento e respetivo problema, e ainda à abordagem para a sua resolução.

No segundo capítulo serão apresentados projetos que foram de alguma forma considerados pertinentes para a realização desta dissertação, quer pelas suas abordagens quer pelos conhecimentos transmitidos.

O terceiro capítulo é referente aos dados utilizados para a realização desta dissertação e respetiva descrição.

Nos capítulos quatro, cinco e seis irão ser implementados diversos métodos numa tentativa de solucionar o objetivo inicial. Para isso serão criadas e descritas detalhadamente várias abordagens manuais, semiautomáticas e automáticas, tendo por base programação em Python, com auxílio de ferramentas para tratamento de texto, algoritmos de aprendizagem automática, métricas para classificação de texto, etc.

Por fim, no sétimo capítulo serão apresentadas considerações finais sobre o trabalho realizado.

Capítulo 2. Revisão da Literatura

Neste capítulo são apresentados vários estudos relacionados com o problema tratado nesta dissertação. Para isso foram feitas várias pesquisas, não só no sentido de aprofundar o conhecimento sobre o tema, mas também para perceber quais os problemas resolvidos, quais os problemas que persistem, quais os problemas que podem ser melhorados, como melhorá-los, e principalmente, que propostas de resolução possam existir para este problema.

A extração da informação de cada CV submetido é um ponto fundamental para a resolução deste problema, podendo ser feita de diversos modos, de acordo com a metodologia empregue. Esta extração pode ser classificada como sendo automática, feita através de algoritmos de aprendizagem automática ou *Web Mining*, ou pode ser feita semi-automaticamente, com recurso a expressões regulares ou com recurso a pesquisa feita por expressões "criadas à mão" (*Hand-Crafted Features*).

Para uma extração automática da informação contida nos currículos são maioritariamente usados algoritmos como *Support Vector Machines* (SVM) e *Hidden Markov Models* (HMM) (Faliagka et al., 2014; Yu, Guan, & Zhou, 2005; ZhiXiang, Chuang, Bo, & ZhiQing, 2009). Esta é a abordagem mais comum ao problema visto que a Internet tem sido uma ferramenta fulcral no processo de recrutamento pois o acesso à informação é muito mais agilizado, e, por conseguinte, também o é o processo de oferta e procura de trabalho. Por essa razão as empresas recebem uma vasta gama de informação por parte de candidatos sendo essa informação impossível de ser tratada e processada eficientemente por humanos.

Numa abordagem com recurso a *Web Mining* (Faliagka, Tsakalidis, & Tzimas, 2012) os candidatos preenchem uma candidatura online. Nessa candidatura têm a opção de fazer login com o seu *LinkedIn* e ainda a opção de anexar o seu blog, caso o possuam.

São utilizados 4 critérios para avaliação e extração de informação:

- Educação (em anos académicos)
- Experiência de trabalho (em meses)
- Lealdade (média de permanência em trabalhos anteriores)
- Personalidade.

Caso entrem com o seu *LinkedIn*, o sistema extrai automaticamente toda a informação relevante. Alternativamente, caso um blog seja anexado, uma análise linguística é realizada com o objetivo de recolher informação sobre a personalidade do candidato.

Esta é uma vertente de avaliação, que em condições habituais, seria apenas possível de ser observada presencialmente, uma vez que não é possível os recrutadores analisarem a personalidade de uma pessoa apenas pelo seu currículo. Deste modo, esta abordagem contribui para uma maior redução de custos – tempo e dinheiro – ao evitar a fase presencial para uma possível análise da personalidade do candidato.

A abordagem proposta para a análise linguística recorre a *Web Mining*, tendo as seguintes como principais características:

- *Positive emotion words and score count* – Através de *Linguistic Inquiry and Word Count* (LIWC) – quanto mais alto é o score obtido, mais positiva, a nível emocional, é a pessoa.
- *Social Orientation Score* – também obtido através de LIWC. Mede o quão frequentemente o candidato utiliza "*social words*" (*buddy, coworker, friend*) e, novamente, quanto maior o score obtido nesta vertente, mais sociável é a pessoa.

De seguida, é calculado o ranking dos candidatos através de uma técnica denominada AHP – *Analytical Hierarchy Process*. Esta técnica é aplicável a problemas multicritério, por outras palavras, diversos critérios são considerados em simultâneo, sendo escolhida a melhor opção de entre todas as combinações possíveis. Note-se que, nesta abordagem, o peso dado pelos recrutadores aos diferentes critérios para a escolha de candidatos é tido em conta. No que respeita a abordagem por métodos de *Web Mining* é referido por (Faliagka et al., 2012) que o teste para 100 candidatos foi bem-sucedido, quer a identificar traços da personalidade do candidato, quer a atribuir um score adequado à sua candidatura. Para obter esta conclusão foi feita uma *cross-validation* por recrutadores experientes, i.e., após as respostas obtidas pela abordagem com recurso a *Web Mining*, essas respostas foram comparadas com as respostas dadas pelos recrutadores. Neste projeto foi conseguida uma atribuição de scores aos candidatos com uma margem de erro mínima, salvo raras exceções sendo elas o recrutamento para posições seniores que requerem qualificações específicas e a experiência na área de candidatura.

Esta é uma abordagem que não só tem em conta o currículo do candidato como algumas das suas competências sociais, no entanto é uma abordagem que não pode ser

adaptada para a resolução deste problema devido ao facto de, para este, apenas serem submetidos os CVs dos candidatos, sem quaisquer requerimentos adicionais.

Em (Kessler, Torres-Moreno, & El-Bèze, 2007) é retratado um caso real de extração de informação a respostas de ofertas de trabalho por e-mail. Este processo começa quando o e-mail é recebido, a linguagem nele contido é identificada e são também analisados ficheiros anexados ao mesmo. Após isso é feita uma filtragem e lematização da informação, para poder representar cada segmento de texto como um vetor e para, seguidamente, através de SVMs, categorizar corretamente esse segmento de modo a saber o assunto presente nesse segmento, i.e., se se refere a salário, descrição do trabalho, etc. Este processo de extração do tema de cada segmento de texto passa de seguida por um "algoritmo corretivo" que valida, ou propõe, uma melhor sequência para o texto, i.e., um documento é esperado que comece com um título, seguido de uma descrição, de uma missão, de um perfil esperado, e por fim uma conclusão. Logo, se esta sequência não é respeitada, o algoritmo entra em ação.

Segundo os autores, de maneira a localizar o tipo de informação que está presente em cada segmento, após a extração, foram utilizadas várias soluções, tais como:

• Salário – foram criadas expressões regulares para encontrar frases como "Salário: de X a Y", "Salário: entre X e Y" ou "Salário: fixo de X, com bónus de Y".

• Local de trabalho – foi criada uma tabela com os campos área, cidade, e departamento, para encontrar a localização de um determinado trabalho, pois muitos dos trabalhos são categorizados por área para ajudar os candidatos na procura.

Experiências preliminares mostram que a categorização dos segmentos sem a posição predefinida dos mesmos, i.e., seguindo uma ordem lógica de estruturação (Título, Descrição, etc.), não é suficiente para os classificar. As SVMs geralmente classificavam bem os segmentos, no entanto não os classificavam a todos. Para isso foi criado uma HMM com 6 estados: *Start – S, Title – 1, Description – 2, Mission – 3, Profile – 4, End – E.*

Ao usar classificação dos segmentos por meio de SVMs alguns segmentos eram mal classificados e, por isso, foi usado um algoritmo do tipo Viterbi (G. D. Forney, 1973). Isto é, as SVMs classificam os segmentos do documento criando assim uma sequência de classes, por exemplo S – 2 – 2 – 1 – 3 – 3 – 4 – E e, de seguida é calculada a probabilidade dessa sequência. Caso não seja provável, entra em ação o processo corretivo que retorna a sequência com erro mínimo e maior probabilidade.

Tal como foi referido anteriormente, a utilização de SVM para a classificação dos segmentos conseguiu bons resultados, no entanto, com a adição de uma HMM, de um algoritmo Viterbi e de um processo corretivo, o processo foi significativamente melhorado. Enquanto as SVMs atingem os 50% de anúncios não reconhecidos, o processo corretivo apenas não reconhece 20%. Por outro lado, termos importantes para a classificação de um segmento de texto por vezes são encontrados na fronteira dos segmentos, sendo categorizados como 2 tópicos e, por isso mesmo, necessitando de correção.

Na abordagem feita por (Yu et al., 2005), a informação contida num currículo é categorizada como sendo informação geral ou detalhada, tratando cada uma delas com uma abordagem diferente. Nesta abordagem é utilizado um método de extração automática dos vários segmentos de um currículo, o modelo utilizado neste estudo consiste na extração dos diferentes blocos de um currículo e na sua classificação utilizando uma HMM para uma primeira triagem e classificação dos vários blocos de texto presentes nos currículos.

Os currículos são segmentados em blocos, usando um HMM, e é aplicado um segundo HMM mas, neste caso, para extrair informação dos blocos já segmentados. Adicionalmente é ainda utilizada, após a primeira segmentação, uma SVM, por forma a retirar informação dos blocos já segmentados. Foram realizados vários testes. Primeiro foi aplicado apenas uma HMM para classificação dos documentos e, de seguida, uma SVM, não fazendo distinção entre o tipo de informação que cada modelo iria classificar. De seguida foram, novamente, aplicados os modelos isolados, mas neste caso fazendo a distinção entre o tipo de informação que iria ser extraída, isto é, utilizando uma HMM para extração da informação educacional, enquanto que a SVM foi utilizada para extração da informação pessoal.

Neste modelo híbrido apresentado por (Yu et al., 2005), foram testados cerca de 1200 currículos chineses. Ao aplicar a HMM para extração da informação e a SVM de modo isolado, os melhores resultados são obtidos sem qualquer divisão da informação com recurso à SVM. No entanto, ao aplicar os mesmos modelos à informação já dividida, os resultados diferem dos anteriores, sendo que a SVM é melhor a classificar o bloco de informação pessoal e a HMM apresenta melhores resultados aquando da classificação da informação referente à educação. Conclui-se, assim, que o modelo híbrido em causa apresenta melhores resultados para os diferentes tipos de informação que é necessário

extrair, quando comparado à utilização de apenas um dos modelos isoladamente para a extração da mesma informação.

(ZhiXiang et al., 2009) apresenta um estudo de extração de informação de currículos através de algoritmos de extração de informação e ainda de expressões regulares. A abordagem seguida por estes autores para a classificação dos segmentos usa modelos manuais e modelos automáticos, classificando os textos em duas classes: Básica e Complexa. A informação pessoal é caracterizada como sendo informação básica, enquanto que, campos que apresentam mais parágrafos por elemento, tais como experiência profissional e experiência educacional, são considerados informação complexa.

Para a extração da informação das referidas classes são utilizados diferentes métodos, de acordo com a complexidade da informação que é necessário extrair. Para extração da informação contida na classe básica são utilizados métodos baseados em regras e também usados métodos automáticos, mais propriamente *"Floating Window"* e "*Squeeze*". O primeiro baseia-se na localização da informação contida após o título, pois em alguns currículos a informação não se encontra seguida ao título, mas sim por baixo. O segundo algoritmo baseia-se na localização da informação presente no texto entre dois títulos. Para a classe complexa são não só utilizados métodos baseados em regras, mas também métodos para a classificação de texto automáticos. De modo a que o CV seja separado corretamente nas várias vertentes pelas quais esta classe é composta, são utilizadas subclasses: uma para a experiência profissional, outra para a educação, etc. Após esta divisão em subclasses são então aplicados métodos baseados em regras para a extração da informação complexa. No entanto, no caso da informação classificada como complexa, se as palavras não corresponderem às expressões regulares, é aplicada uma SVM para classificação do bloco e, só depois, é aplicada de novo a procura através de expressões regulares, e, por conseguinte, a sua extração.

Nesta abordagem para a classificação da informação são também utilizados métodos baseados em regras criadas à mão que, neste caso, representam a correspondência de palavras usando expressões regulares e, ainda, métodos que têm por base HMM. Estes últimos dois métodos são os que apresentam as maiores taxas de sucesso referentes à classificação de texto por segmentos.

Em (Tosik et al., 2015) é apresentada uma utilização mista de modelos automáticos e modelos manuais para processar a informação contida num currículo. Adicionalmente, é

representado um modelo que trata a classificação e extração de informação com recurso a um algoritmo *Conditional Random Files* (CRF), tendo como input para este vários métodos manuais, *Hand-crafted features*, *Word Type*s e *Word Embeddings*.

Para o segmento de texto referente à informação pessoal são extraídos campos para identificação de um candidato, enquanto que, para o segmento de texto referente à experiência profissional, são retirados apenas três campos: Tipo de trabalho, Duração e Empresa e localização. Numa primeira fase, para a criação das *words embeddings*, é usada a ferramenta *word2vec* (W. Ling, C. Dyer, AW. Black, 2005); para a criação de *word types* é criado um vetor que contém todas as palavras que surgem mais do que duas vezes no documento e, por fim, para a criação de características, são utilizadas técnicas simples de forma a melhorar ortograficamente o texto, para de seguida agrupar as 200 palavras mais frequentes contidas nos documentos, também elas num vetor. De seguida foram criados cinco modelos com base no algoritmo CRF para testar quais seriam os inputs que dariam melhores resultados para este estudo. Para isso foram dados como inputs: *hand-crafted features*, *word types*, *word embeddings*, isoladamente, e ainda a junção de *word types* com *hand-crafted features* e de *word embeddings* com *hand-crafted features*. Os inputs referidos foram utilizados quer para o bloco da informação pessoal, quer para o bloco da experiência profissional. Pela análise dos resultados obtidos foi concluído que utilizar apenas *word types* ou *word embeddings* para as classificações dos blocos não seria proveitoso, apresentando uma performance mais baixa que as *hand-crafted features*. No entanto, ao combinar estes dois inputs com *hand-crafted features*, a performance do modelo aumenta significativamente, fazendo assim uma melhor categorização do bloco.

A utilização conjunta de modelos automáticos e modelos manuais pode ser uma abordagem bastante coesa para a classificação do texto de um CV mais precisamente dos blocos relacionados com informação pessoal e com a experiência profissional, tratando o problema como se de um problema de *Name Entity Recognition* se tratasse (Tjong, Sang, & Meulder, 2003).

Capítulo 3. Descrição dos Dados

Para a realização desta dissertação foram analisados vários currículos facultados pela Siemens, enviados por candidatos reais para análise e possível recrutamento para cargos reais. Todos os currículos foram retirados de uma base de dados (4Success) utilizada pelo departamento de Gestão de Recursos Humanos da empresa em questão. Esta base de dados possui todos os cargos da empresa para os quais foram abertas vagas, onde, para cada cargo, existem vários currículos que foram ou irão ser analisados pelos recursos humanos e, por conseguinte, serão aceites ou declinados consoante a qualidade do candidato.

Foram escolhidos para este estudo cerca de 20 currículos de 25 cargos diferentes, perfazendo 500 currículos. Estes cargos e respetivos currículos foram escolhidos devido ao facto de terem sido previamente analisados pelos recursos humanos de cada departamento e, por isso, a cada currículo foi atribuída uma anotação de acordo com a qualidade do candidato, podendo, esta anotação, ser classificada como "Bom Candidato", "Mau Candidato" ou "Incerto", facilitando assim uma possível validação para medidas de desempenho do trabalho desenvolvido.

Para cada grupo de currículos pertencentes a cada cargo foi também recolhido o respetivo anúncio de oferta de trabalho, que será denominado, no âmbito deste trabalho, "Job ID". O conteúdo desta Job ID difere de departamento para departamento, no entanto a sua estrutura mantém-se, sendo esta composta por qual o cargo em questão, keywords descritivas acerca do cargo, breve descrição da empresa, o que a empresa procura num candidato, quais as responsabilidades do candidato, quais as qualificações necessárias e por fim qualificações preferenciais.

Foi também facultado, pelos Recursos Humanos de cada departamento, um conjunto de CVs não analisados, num total de 800 CVs não anotados e sem referência quanto à sua qualidade para preencher um cargo de um certo departamento.

Para início de desenvolvimento dos trabalhos, todos os CVs foram analisados quanto à sua estrutura, tipo de ficheiro e conteúdo, de maneira a fazer uma triagem de quais os CVs que poderiam ser utilizados (Tabela 1), juntando estes aos 500 já considerados como utilizáveis.

Tabela 1 – Triagem dos CVs não anotados

Categoria de CVs	Nº de CVs
Utilizáveis	487
Não Utilizáveis	313
Total	800

Foram maioritariamente considerados como não utilizáveis CVs que estão organizados por colunas (Figura 1), CVs que são apresentados como ficheiro de imagem (Figura 2), e ainda CVs em que o texto não é selecionável.

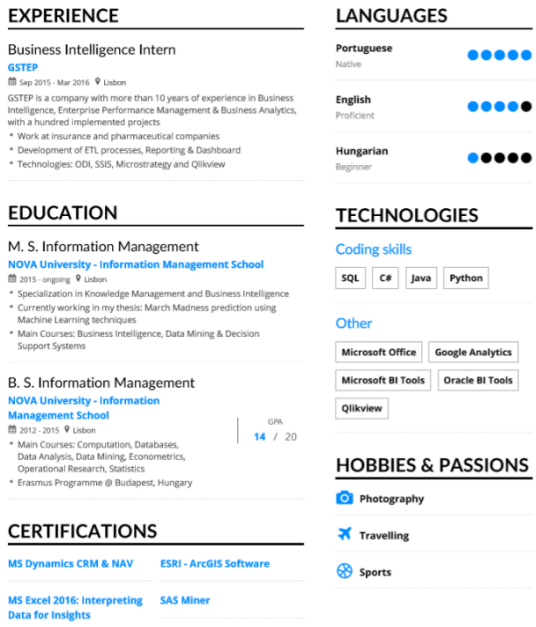


Figura 1 – Exemplo de CV organizado por colunas

Figura 2 – Exemplo de CV apresentado por imagem

Após a análise de todos os conjuntos de CVs foi concluído que o Corpus seria composto pelos dois tipos (os CVs anotados e os não anotados) utilizáveis. Para conjunto de teste foi escolhido o dos CVs anotados e como conjunto de treino o conjunto dos não anotados, criando assim um conjunto de dados com um total de 987 CVs (Tabela 2).

Tabela 2 – Tabela ilustrativa dos CVs que constituem o Corpus.

Conjunto de CVs	Nº de CVs
Utilizáveis - Não Anotados	487
Utilizáveis - Anotados	500
Total	987

3.1. Pré-processamento dos Dados

Como foi apresentado no Capítulo 3 desta dissertação a primeira fase deste trabalho consistiu em garantir que todos os documentos estivessem no mesmo formato, de maneira a serem todos trabalhados igualmente. Assim, a primeira tarefa foi a de criar um conversor para a uniformização dos dados, tendo em conta que não poderia ser perdida informação aquando da conversão.

Dado que cerca de 90% dos dados se encontrarem em formato PDF, era imperativo criar um conversor que mantivesse a integridade e formatação dos dados, mas que convertesse os documentos num formato operacional. Após vários testes com vários conversores, foi criado um conversor de PDF para o formato mais simples de texto, TXT, utilizando uma ferramenta *opensource pdf2txt*[1]. Foi testado primeiro para um só documento e, de seguida foi criada uma *batch* (ferramenta para execução de comandos), através da Shell do Windows, para que a conversão fosse efetuada para todos os ficheiros PDF – CVs (Figura 3).

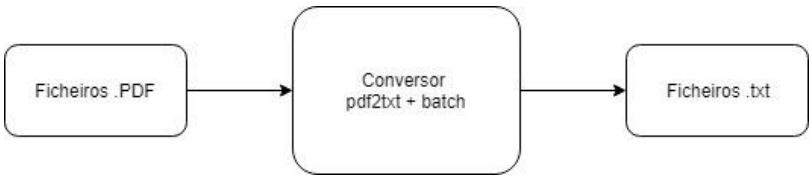


Figura 3 – Conversor de ficheiros .PDF para .txt

Os restantes 10% dos CVs estavam em formato DOCX, pelo que seria necessário um conversor DOCX para TXT. Para isso, foi criado um conversor DOCX, que converte todos os ficheiros para formato PDF, sendo posteriormente convertidos para texto pelo conversor criado anteriormente (Figura 4).

[1] pdf2txt - https://www.xpdfreader.com/pdftotext-man.html.

Figura 4 – Conversor final de ficheiros .docx para .txt

Foi então necessário visualizar os diferentes conteúdos dos documentos constituintes do Corpus. Verificou-se que estavam presentes vários modelos de organização e apresentação de CVs, tais como: *Europass*, modelos criados à mão pelo candidato, entre outros, o que tornou mais complexa a tarefa de estruturação dos dados neles presentes.

Adicionalmente, definiram-se grandes categorias de informação, optando-se por escolher 4 categorias nas quais os dados se podem repartir:

- Informação Pessoal
- Experiência Profissional
- Educação
- Outros

São classificados como "Informação Pessoal" todos os dados pessoais de um candidato tais como nome, data de nascimento, e-mail, número de telefone e morada.

Na categoria de "Experiência Profissional" são inseridos todos os dados referentes a cargos que um candidato tenha ocupado, bem como as respetivas empresas e a duração desta atividade.

A categoria referente à "Educação" contém todos os dados relativos à formação do candidato, duração dessa formação e ainda o(s) estabelecimento(s) que frequentou.

Relativamente à categoria "Outros", trata-se da categoria mais ampla das quatro consideradas. Nesta, inserem-se diversos assuntos que não se enquadram em nenhuma das outras categorias, não assumindo tão pouco um peso tão notório no processo de recrutamento. Entre outras possibilidades, esta categoria inclui aptidões e competências pessoais e sociais, línguas dominadas e carta de condução.

A necessidade da repartição de um CV nestes quatro grupos prende-se com o facto de, desta forma, ser mais fácil segmentar a informação por categorias para que essa informação seja trabalhada de maneira independente, dado que todas contêm diferente informação.

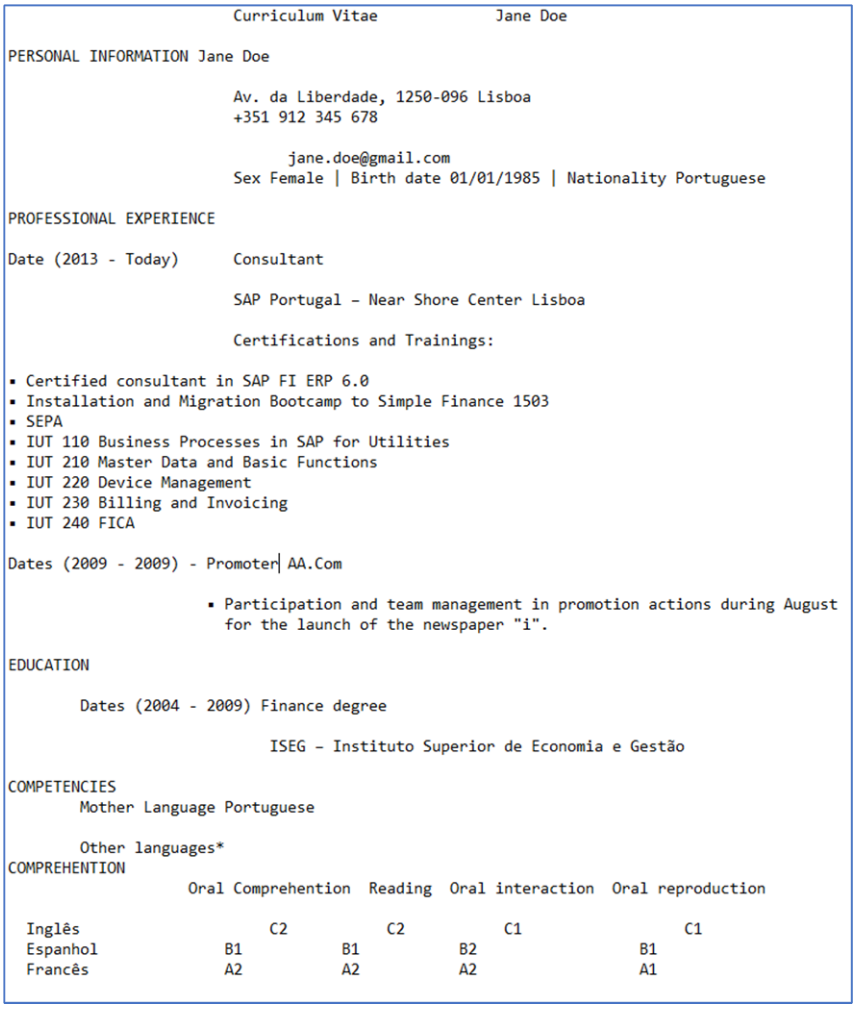
Capítulo 4. Recuperação de Informação

Como referido no ponto 3.1) a divisão de um CV pelas quatro categorias é uma forma de facilitar a recolha de informação presente em cada uma delas para, posteriormente, esta ser trabalhada. No seguimento deste capítulo irão ser implementadas técnicas predominantemente focadas na divisão de um CV em blocos de informação categorizada. Serão apresentadas de seguida várias abordagens por ordem de simplicidade, i.e., incrementando, a cada abordagem, o nível de complexidade de cada uma, com o objetivo de melhorar sempre a que se precede.

Numa primeira abordagem serão utilizadas listas de palavras para a divisão de um CV por categorias. De seguida, será feita uma abordagem baseada em dicionários com o mesmo intuito da anterior. Numa terceira abordagem irá ser apresentada uma versão melhorada da segunda, com recurso a *TF-IDFVectorizer*. A quarta e a quinta abordagens têm como objetivo a divisão de um CV em blocos definidos pelos títulos presentes no mesmo, sendo que, na última abordagem, com o intuito de melhorar a antecedente, recorreu-se ao uso de expressões regulares.

4.1. Information Retrieval com recurso a listas de palavras

Numa primeira abordagem à resolução deste problema tentou-se criar uma ferramenta simples para resolução do mesmo, tendo esta ferramenta por base a identificação das palavras predominantes contidas em cada uma das categorias previamente descritas no Capítulo 3 e tentando, através delas, linha a linha, classificar a que categoria cada linha de um determinado CV pertence de acordo com as palavras contidas nessa linha. Para isso foi necessário fazer o levantamento de todas as palavras contidas em todas as categorias e, por essa razão, foi necessário dividir manualmente, por categorias, parte dos CVs do conjunto de treino.



Curriculum Vitae Jane Doe

PERSONAL INFORMATION Jane Doe

Av. da Liberdade, 1250-096 Lisboa
+351 912 345 678

jane.doe@gmail.com
Sex Female | Birth date 01/01/1985 | Nationality Portuguese

PROFESSIONAL EXPERIENCE

Date (2013 - Today) Consultant

SAP Portugal – Near Shore Center Lisboa

Certifications and Trainings:

- Certified consultant in SAP FI ERP 6.0
- Installation and Migration Bootcamp to Simple Finance 1503
- SEPA
- IUT 110 Business Processes in SAP for Utilities
- IUT 210 Master Data and Basic Functions
- IUT 220 Device Management
- IUT 230 Billing and Invoicing
- IUT 240 FICA

Dates (2009 - 2009) - Promoter AA.Com

- Participation and team management in promotion actions during August for the launch of the newspaper "i".

EDUCATION

Dates (2004 - 2009) Finance degree

ISEG – Instituto Superior de Economia e Gestão

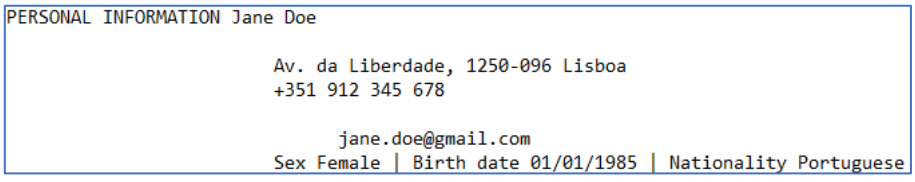
COMPETENCIES
Mother Language Portuguese

Other languages*
COMPREHENTION

	Oral Comprehention	Reading	Oral interaction	Oral reproduction
Inglês	C2	C2	C1	C1
Espanhol	B1	B1	B2	B1
Francês	A2	A2	A2	A1

Figura 5 – Exemplo de um CV antes de ser dividido manualmente por categorias.

Deste modo, foram criados, para cada documento, quatro novos documentos, derivados do primeiro, de acordo com a sua categoria: Informação Pessoal, Experiência Profissional e outros. Da Figura 6 à Figura 9 podem ser visualizados os novos documentos criados a partir do CV do exemplo ilustrado na Figura 5.

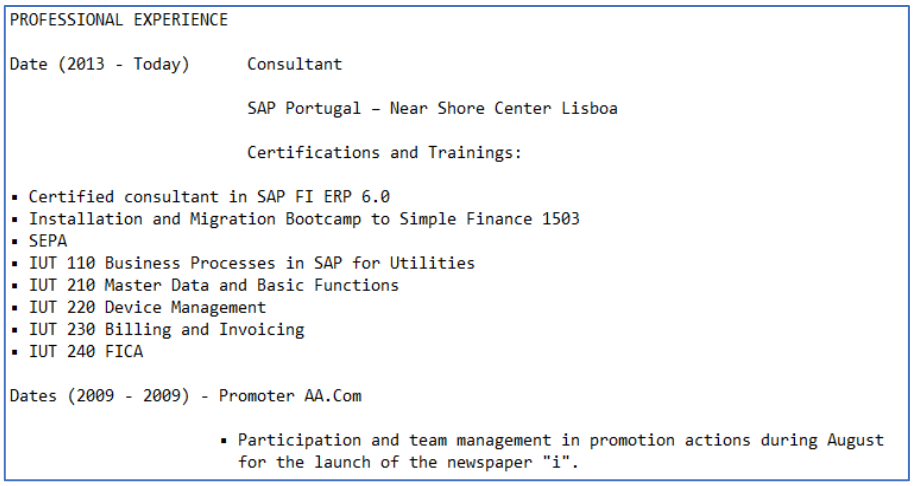


PERSONAL INFORMATION Jane Doe

Av. da Liberdade, 1250-096 Lisboa
+351 912 345 678

jane.doe@gmail.com
Sex Female | Birth date 01/01/1985 | Nationality Portuguese

Figura 6 – CV exemplo depois de dividido – Categoria Informação Pessoal



PROFESSIONAL EXPERIENCE

Date (2013 - Today) Consultant

SAP Portugal – Near Shore Center Lisboa

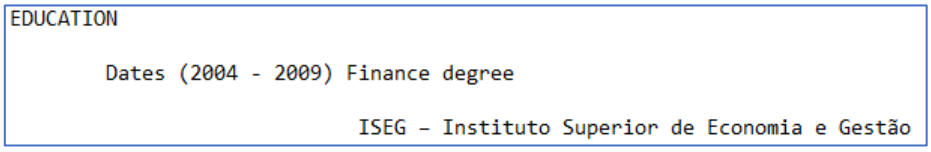
Certifications and Trainings:

- Certified consultant in SAP FI ERP 6.0
- Installation and Migration Bootcamp to Simple Finance 1503
- SEPA
- IUT 110 Business Processes in SAP for Utilities
- IUT 210 Master Data and Basic Functions
- IUT 220 Device Management
- IUT 230 Billing and Invoicing
- IUT 240 FICA

Dates (2009 - 2009) - Promoter AA.Com

- Participation and team management in promotion actions during August for the launch of the newspaper "i".

Figura 7 – CV exemplo depois de dividido – Categoria Experiência Profissional

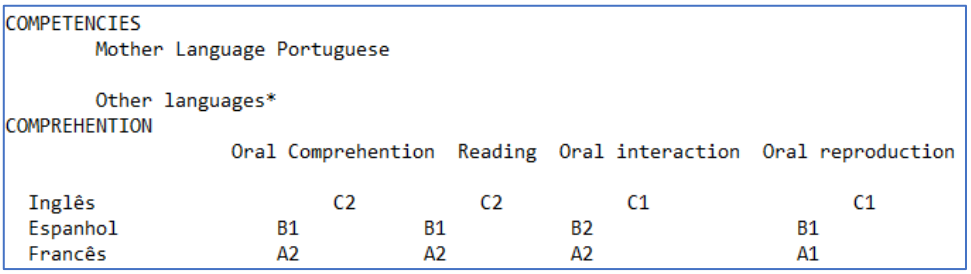


EDUCATION

Dates (2004 - 2009) Finance degree

ISEG – Instituto Superior de Economia e Gestão

Figura 8 – CV exemplo depois de dividido – Categoria Educação



COMPETENCIES

Mother Language Portuguese

Other languages*

COMPREHENTION

	Oral Comprehention	Reading	Oral interaction	Oral reproduction
Inglês	C2	C2	C1	C1
Espanhol	B1	B1	B2	B1
Francês	A2	A2	A2	A1

Figura 9 – CV exemplo depois de dividido – Categoria Outros

Depois de feita esta divisão prosseguiu-se então à contagem de todas as palavras contidas em cada categoria, sendo criado para este efeito um excerto de código em Python. Nesta fase foram criadas quatro diretorias, uma por categoria, para que fosse possível a criação de um código que permitisse a contagem de todas as palavras de todos os documentos pertencentes a cada uma das diretorias.

De seguida já com a ajuda do código anteriormente criado, obteve-se uma lista de todas as palavras para cada uma das categorias. Apresenta-se em baixo, a título de exemplo, a lista de palavras para a categoria Informação Pessoal:

['Doe', 'Lisboa', 'gmail', 'date', 'Nationality', 'Portuguese', 'Personal', 'information', 'First', 'name', 'Surname', 'John', 'Address', 'Mobile', 'mail', 'zhgomes', 'Date', 'birth', 'Dec', 'Gender', 'Male', 'French', 'daughters', 'address', 'Alfort', 'avenue', 'Cedex', 'Phone', 'Mail', 'hotmail', 'software', 'experience', 'test', 'automation', 'Joana', 'Lancaster', 'University', 'fontes', 'joana', 'Porto', 'Rua', 'Portugal', 'Engineering', 'world', 'well', 'production', 'logging', 'many', 'SPE', 'member', 'Nome', 'Marques', 'Data', 'Nascimento', 'Abril', 'Pessoal', 'completo', 'Castro', 'Azevedo', 'Pereira', 'nascimento', 'Nacionalidade', 'Portuguesa', 'pessoal', 'Morada', 'Telefone', 'Santos', 'Estrada', 'Correio', 'Sexo', 'Masculino', 'correio', 'andre', 'profissional', 'JOANA', 'iseg', 'utl', 'NOME', 'NASCIMENTO', 'NACIONALIDADE', 'portuguesa', 'lido', 'falado', 'Social', 'Freire', 'Professor', 'Local', 'Pedreira', 'ISCTE', 'Avenida', 'Telf', 'freire', 'DADOS', 'PESSOAIS', 'isadora', 'castro', 'Buanar', 'Samuel', 'Estado', 'Dias', 'Feteira']

Como é possível observar, as listas contêm uma grande quantidade de palavras, tal como seria de esperar. Muitas destas palavras não são relevantes para definir uma categoria, pelo que foi necessário refazer o código para efetuar um pré-processamento dos documentos. Nesta fase foram então implementadas duas novas características sendo uma delas *"stopwords*" quer portuguesas quer inglesas, i.e., palavras irrelevantes para este contexto, como por exemplo "de, a, o , que, e, do, da, em , um, para", para português, ou "i, me, you, the, it's, is, are, am, and, but", para o inglês. A outra característica implementada para o pré-processamento dos dados foi um contador de palavras para que apenas as que são utilizadas mais do que "X" vezes fossem adicionadas à lista. Neste caso "X" é determinado como sendo a frequência mínima de uma palavra numa diretoria de documentos de uma dada categoria para que seja integrada na lista de palavras respeitante a categoria.

Após vários testes com as diversas listas conclui-se que a frequência mínima que uma palavra deveria ter para ser inserida numa lista poderia variar consoante as respetivas categorias, visto que, a frequência mínima para as palavras da lista denominada "Informação Pessoal" vai ter necessariamente que ser menor do que a frequência mínima de qualquer outra das três listas. De facto, e tal como o seu nome indica é uma categoria em que a informação é muito pessoal e, devido a esse fator, varia sempre de candidato para candidato (Tabela 3), levando a que haja uma variabilidade grande nas palavras usadas na descrição.

Para a lista de palavras pertencentes à categoria "Informação Pessoal", a frequência mínima de repetição de palavras foi escolhida como sendo de cinco (5), pois foi com esse

valor que foram obtidos os melhores resultados, i.e., foi com este valor que as listas aparecem compostas por palavras que descrevem bem uma categoria. Note-se que, o nome da mesma pode não ser conhecido, como é possível observar pelo exemplo da lista a seguir demonstrada:

['Lisboa', 'gmail', 'mail', 'birth', 'Joana', 'Rua', 'Portugal', 'Nome', 'Data', 'nascimento', 'Nacionalidade', 'Portuguesa']

No entanto, o tipo de palavras reconhecidas são claramente pertencentes à categoria em causa.

Seguindo a mesma ordem de ideias, a lista de palavras que caracterizam a categoria "Educação", mantendo a mesma frequência mínima iria resultar numa lista com muitas palavras redundantes para o contexto de educação. Assim sendo, foi necessário alterar o valor da frequência mínima para esta categoria em específico, neste caso para quinze (15), valor com o qual foram obtidos os melhores resultados, passando de uma lista com inúmeras palavras para uma lista com apenas vinte e três, todas boas caracterizadores da categoria "Educação".

O mesmo procedimento foi efetuado para "Experiência Profissional", adequando os valores da frequência mínima para a criação das respetivas listas com o menor número possível de palavras irrelevantes. O valor escolhido para a frequência mínima da lista que caracteriza a categoria "Experiência Profissional" foi de trinta (30).

Por último, para a lista das palavras caracterizadoras da categoria "Outros" a frequência mínima foi de certa forma irrelevante pois é uma categoria muito generalista. A maior parte dos CVs contém nesta categoria informação adicional sobre o candidato que não pertence a nenhuma outra categoria, como que uma junção de vários temas num só, tais como hobbies, voluntariados, etc., o que faz com que não seja possível atribuir-lhe um tema em concreto, e por conseguinte, um conjunto de palavras que caracterizem esse tema. A Tabela 3 indica as diferenças em tamanho de listas quando usando ou não usando limites de frequência para inclusão de palavras.

Tabela 3 – Tabela ilustrativa do número de palavras por categoria com e sem filtro de frequência mínima

Categoria	Nº Palavras sem filtro de frequência mínima	Nº de Palavras com filtro de frequência mínima
Informação Pessoal	101	12
Educação	679	23
Experiência Profissional	910	43
Outros	744	-

Finalizadas as listas de palavras para cada uma das categorias colocou-se em prática a primeira tentativa de solução deste problema que, tal como descrito anteriormente, consiste em tentar classificar uma linha de um CV, categorizando-a, com base nas palavras que estão presentes na mesma. Deste modo, foi criado código para percorrer um CV, lendo-o linha a linha, e, se existirem uma ou mais palavras nessa linha que estejam presentes numa das listas de palavras associadas às quatro categorias, então essa linha é categorizada como sendo dessa(s) categoria(s). O problema desta abordagem simplista é o facto de uma palavra poder estar presente em todas as listas de palavras das quatro categorias o que faz com que todas as linhas que contêm essa palavra sejam classificadas quatro vezes, uma por cada categoria.

```
INFORMAÇÃO PESSOAL: SAP Portugal – Near Shore Center Lisboa
INFORMAÇÃO PESSOAL: ▪ IUT 210 Master Data and Basic Functions
```

Figura 10 – Exemplo de classificação errada de linhas – Informação Pessoal

```
EXPERIÊNCIA: Av. da Liberdade, 1250-096 Lisboa
EXPERIÊNCIA : jane.doe@gmail.com
```

Figura 11 – Exemplo de classificação errada de linhas – Experiência Profissional

```
EDUCAÇÃO: Av. da Liberdade, 1250-096 Lisboa
```

Figura 12 – Exemplo de classificação errada de linhas – Educação

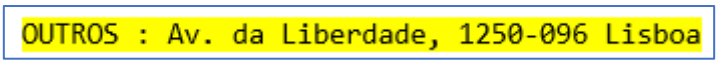


Figura 13 – Exemplo de classificação errada de linhas – Outros

Por exemplo, a linha "Av. Da Liberdade, 1250-096 Lisboa", do CV da Figura 5, aparece classificada nas 4 categorias. Esta multi-classificação deve-se ao facto de a palavra "Lisboa" estar contida em todas as listas de palavras associadas às categorias. Assim, categoriza uma linha que pertence apenas a "Informação Pessoal" como sendo simultaneamente das restantes categorias. Este tipo de situações tornam esta abordagem inviável para um problema complexo como este.

4.2. Information Retrieval com recurso a dicionários

Após a anterior abordagem, e devido ao facto desta se ter mostrado ineficiente para a classificação de categorias de informação, tentou-se implementar uma nova abordagem para classificação de linhas de um CV, desta vez, com base em dicionários.

Dicionários em Python são coleções de dados constituídos por um par chave-valor (Drake, 2002). Para esta abordagem o par chave-valor constituinte do dicionário irá ser: palavra e número de vezes que essa palavra ocorre na categoria. Logo o dicionário irá conter todas as palavras e as vezes que elas ocorrem por categoria.

Esta abordagem, tal como a anterior, consiste em classificar um CV, linha a linha, categorizando-as, de entre as quatro categorias de acordo com o grupo de palavras que constituí essa linha.

Também nesta implementação é necessário isolar as categorias para obter, neste caso, um dicionário fidedigno de palavras caracterizadoras. Para isso em vez de os CVs serem divididos manualmente em quatro novos documentos, como anteriormente, parte dos CVs foram anotados para servirem como conjunto de treino. Estas anotações consistem simplesmente em apontar em cada currículo as "fronteiras" entre categorias dentro de um CV para que, aquando da implementação do código, a leitura do CV por categorias seja mais fácil (Figura 14).

Foi implementado um programa que lê cada bloco de informação por categoria entre anotações. Se, porventura, uma anotação foi escrita erradamente, i.e. diferente de

"PI","EDU",EXP" ou "OT", ou num sítio que não o correto é dada uma mensagem de erro aquando da leitura do CV.

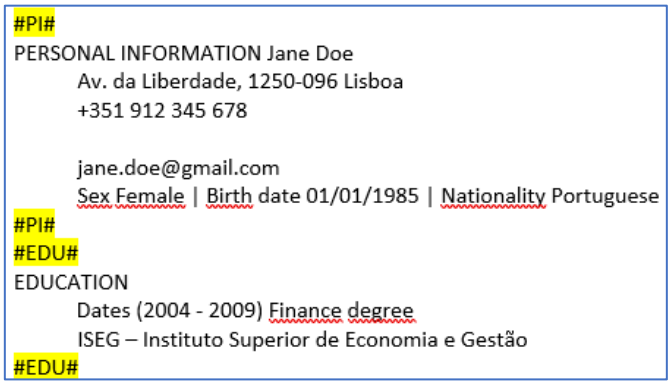


Figura 14 – Exemplo de CV anotado nas fronteiras de cada segmento

Após a anotação de todos os CVs de treino foi necessário criar quatro dicionários, um por categoria, para que cada categoria tenha novamente o seu grupo de palavras. Já com os dicionários criados foi observado que, tal como no ponto 4.1) 4.1. estes iriam ter uma coleção de palavras muito extensa pelo que foi necessário mais uma vez recorrer a um filtro numérico do número de ocorrências que uma palavra necessita de ter para ser inserida no dicionário (Tabela 4).

Tabela 4 – Número de palavras constituintes de cada dicionário sem qualquer tratamento

Categoria	Nº de Palavras
Informação Pessoal	322
Educação	395
Experiência Profissional	664
Outros	2213

Para esta implementação foram também removidas "*stopwords*" quer em inglês quer em português, porém foi alterado o modo como são filtradas as palavras, em vez de uma palavra ter que ocorrer "X" vezes para ser inserida no dicionário, o utilizador escolhe o número de palavras mais frequentes que quer utilizar para criar o mesmo. No exemplo usado, foi escolhido o valor cinco. Os dicionários foram criados usando as cinco palavras mais frequentes naquela categoria (Tabela 5).

Tabela 5 – Exemplo de dicionários criados com as cinco palavras mais frequentes

Categoria	Palavra	Ocorrência
Informação Pessoal	E-mail	8
	Birth	6
	Date	6
	Personal	5
	Address	4
Educação	University	17
	Research	11
	Education	8
	Communication	7
	Paris	6
Experiência Profissional	Member	28
	President	16
	Veterinary	14
	Since	14
	French	13
Outros	Evolution	57
	Biology	43
	C	36
	Language	24
	Portugal	23

Para a classificação de linhas utilizando este procedimento foi ainda necessário criar um quinto dicionário auxiliar com os pares categoria-contador. Por essa razão foram criados quatro contadores, inicializados a zero, para contarem quantas palavras do texto foram identificadas como pertencentes a cada uma das categorias.

Com a criação dos dicionários e a implementação da filtragem de palavras foi possível classificar as linhas de um CV. O modo como a classificação opera é idêntico ao referido no ponto 4.1), o CV é lido, linha a linha, e as palavras que estão presentes na linha que está a ser lida são comparadas com as palavras presentes nos quatro dicionários. Quando há interseção, é incrementado o contador da categoria respetiva no quinto dicionário. No fim da leitura do CV, cada linha irá ter um dicionário com os seus pares categoria-

contador, e, por conseguinte, o contador com maior valor será o que dita a categoria dessa linha (Figura 15).

```
{'PI': 3, 'WE': 0, 'ED': 0, 'OT': 0}
Linha : jane.doe@gmail.com - Categoria : PI

{'PI': 0, 'WE': 5, 'ED': 3, 'OT': 2}
Linha :  Teaching Computer Science II (in java) to undergraduate students - Categoria : WE
```

Figura 15 – Output exemplo de linhas corretamente classificadas

No entanto, esta abordagem de classificação de linhas apresenta também ela lacunas, similares às da abordagem da qual deriva, como por exemplo palavras que estão presentes numa linha de uma certa categoria mas que ocorrem maioritariamente noutras categorias (Figura 16).

```
{'PI': 0, 'WE': 3, 'ED': 2, 'OT': 0}
Linha :  Dates (2004 - 2009) Finance degree - Categoria : ED
Classificada como : WE


{'PI': 2, 'WE': 0, 'ED': 1, 'OT': 1}
Linha : Personal skills and competences - Categoria : OT
Classificada como : PI
```

Figura 16 – Outputs exemplo de linhas erradamente classificadas

Esta abordagem continua a ser pouco ortodoxa para o tratamento deste problema, pois para esta forma de abordar a resolução deste problema ter sucesso teriam que existir dicionários perfeitos para cada categoria, sem qualquer intersecção entre eles.

4.3. Information Retrieval com recurso a dicionários e *TF-IDFVectorizer*

Como foi possível observar pelas abordagens nos pontos 4.1) e 4.2) a tentativa de classificação de linhas utilizando apenas métodos simples de tratamento de palavras, não são abordagens que acrescentem uma mais valia para a resolução deste problema.

Por essa razão, nesta fase, implementou-se uma variação da abordagem anterior. Nesta abordagem a divisão dos CVs por categorias é feita como no ponto 4.2), sendo utilizados, também aqui, dicionários. No entanto, a inserção das palavras nos mesmos é feita de maneira diferente.

O método pelo qual as palavras são inseridas nos dicionários é denominado por *TF-IDFVectorizer* (Pedregosa & Varoquaux, 2011). Este método deriva do método *TF-IDF*, onde TF se refere à frequência dos termos num documento ("*Term-Frequency*"), e IDF à frequência inversa dos documentos ("*Inverse Document Frequency*"), i.e., com que frequência uma palavra ocorre no conjunto de documentos. Resumidamente é um método que pretende calcular a importância de uma palavra num documento de entre uma coleção de documentos (Joachims, 1996; Medina & Ramon, 2015).

Para esta abordagem são utilizados, novamente, os CVs divididos por categoria, para que com recurso ao *TF-IDFVectorizer*, este método aprenda todas as palavras relacionadas com uma categoria, criando com elas um dicionário denominado por vocabulário no qual o par chave-valor será palavra do dicionário e índex da palavra no dicionário respetivamente (Tabela 6), para de seguida calcular o *TF-IDF* de cada uma dessas palavras e com isso classificar uma linha corretamente.

Tabela 6 – Exemplo ilustrativo de dicionário utilizando TF-IDFVectorizer

Categoria	Palavra	Índex no Dicionário
Informação Pessoal	Street	296
	Jane	190
	Doe	155
	Info	138
	Summary	299

O objetivo da implementação deste método é conhecer quais as palavras mais importantes para cada categoria numa linha e, por essa razão, pesarem mais na classificação de uma linha. Quer isto dizer que essas palavras podem estar presentes noutra categoria mas não tendo tanto peso, não interferem no processo de classificação. Com recurso a este método os vocabulários criados para cada categoria podem não só conter coleções de palavras isoladas (*"Unigrams"*), mas também coleções de duas ou mais palavras ("*Bigrams"*, *"Trigrams"*, etc."), denominados por *"n-grams"*(Cavnar, 1994).

A utilização de *"n-grams"* para efeitos de classificação do texto pode ser uma mais valia pois muitas das vezes existem palavras que singularmente não representam qualquer ligação com uma categoria, mas que agrupadas com outra(s) palavra(s), devido à

frequência com que aparecem juntas, podem trazer relevância para o contexto em causa (Tabela 7).

Tabela 7 – Exemplo de n-grams

	Unigrama	Bigrama	Trigram
Palavra(s)	Acquired	Acquired immune	Acquired immune deficency

Para se conseguir tirar proveito da utilização de "*n-grams*" é necessário um pré-processamento de todo o conteúdo dos CVs. Para isso foram removidas "stopwords", acentos, caracteres especiais e pontuações. De seguida todo esse conteúdo foi "*tokenizado*"[2],i.e., separar uma frase pelas palavras que a constituem com recurso a um separador, palavras essas denominadas por "*tokens*". Neste caso, o separador considerado foi apenas um espaço (Tabela 8) e a separação foi feita automaticamente pelo *TF-IDFVectorizer*.

Tabela 8 – Exemplo de linha tokenizada

Linha	Esta linha é o exemplo de uma linha *tokenizada*								
Tokens	Esta	linha	é	o	exemplo	de	uma	linha	*tokenizada*

Ao contrário das anteriores abordagens não foi necessário implementar uma filtragem de palavras através da sua frequência mínima pois, tal como já foi dito previamente, o método em causa utiliza todas as palavras existentes no vocabulário, mas consoante a sua frequência atribui-lhes um peso maior ou menor quanto à sua importância.

Iniciada a fase de criação dos dicionários para cada categoria decidiu-se então testar o tamanho do dicionário quanto à utilização de "unigramas", "bigramas" e "trigramas". Como podemos observar pela Tabela 9 as categorias cujos vocabulários possuem maior número de palavras são as que correspondem às categorias "Experiência Profissional" e a "Outros". Quanto ao vocabulário que corresponde à "Experiência Profissional" seria de esperar este resultado pois é nesta categoria que os candidatos elaboram mais a sua informação. Quanto à categoria de "Outros" o tamanho do vocabulário é o maior de entre todas as categorias pois, aquando da divisão dos CVs por categoria, toda a informação

[2] (https://www.techopedia.com/definition/13698/tokenization)

que não se encaixava nas categorias acima era anexada nesta, fazendo assim com que esta seja a categoria com mais informação entre todas. Quanto ao aumento do número de palavras quando são implementados "bigramas" e "trigramas" era expectável devido ao facto do *TF-IDFVectorizer* agrupar, por exemplo, dois "unigramas" criando um "bigrama", sendo este adicionado ao vocabulário como sendo uma nova palavra.

Tabela 9 – Comparação de tamanho dos dicionários com diferentes n-gramas

	Informação Pessoal	Educação	Experiência Profissional	Outros
Unigramas	331	397	697	2232
Bigramas	779	955	1871	6623
Trigramas	1242	1538	3162	11597

Já com os vocabulários criados e com o *TF-IDFVectorizer* a atuar, para classificar as linhas de um CV. Para esta classificação, tal como foi descrito anteriormente, foi utilizado o vocabulário de cada uma das categorias e o respetivo *TF-IDF* de cada uma das palavras desse vocabulário. O CV é lido linha a linha e são analisadas as palavras que compõem essa linha. É feita uma comparação, palavra a palavra, com o conteúdo de todos os vocabulários. Se uma ou mais palavras estiverem contidas em um, ou mais vocabulários, é devolvido como resultado o valor de *TF-IDF* de cada uma delas, por categoria. Por fim, a linha é classificada como sendo de uma categoria se o valor da soma dos *TF-IDF* de todas as palavras da linha for o maior de entre as quatro.

```
Linha : Sex Female | Birth date 01/01/1985 | Nationality Portuguese
Categoria : PI

Correspondência PI :
        (0, 1093)       0.2864252001675413
        (0, 1092)       0.2864252001675413
        (0, 1091)       0.2864252001675413
        (0, 950)        0.2243292118704479
        (0, 822)        0.2243292118704479
        (0, 817)        0.1622332235733545
        (0, 399)        0.2864252001675413
        (0, 398)        0.2864252001675413
        (0, 397)        0.2864252001675413
        (0, 247)        0.2864252001675413
        (0, 246)        0.2864252001675413
        (0, 240)        0.1622332235733545
        (0, 102)        0.2864252001675413
        (0, 101)        0.2864252001675413
        (0, 100)        0.1622332235733545
Correspondência WE :
        (0, 2198)       0.7904489597281997
        (0, 682)        0.6125279112535257
Correspondência ED :
        (0, 946)        1.0
Correspondência OT :
        (0, 8645)       0.35566114128921
        (0, 2590)       0.6608725870305322
        (0, 1116)       0.6608725870305322
```

Figura 17 – Exemplo de classificação através de TF-IDFVectorizer

Na Figura 17 podemos ver que as correspondências são descritas pelo índice da palavra no vocabulário correspondente seguidas do seu valor *TF-IDF* (ex. 1093 é o índice da palavra 'sex female', neste caso um "bigrama", com o valor *TF-IDF* de 0.2864252001675413).

Tendo mais uma vez como exemplo a Figura 17 é possível observar que a linha a ser classificada pertencia a "Informação Pessoal". Após a classificação automática, quinze palavras da linha corresponderam às palavras pertencentes ao vocabulário dessa mesma categoria, duas palavras corresponderam ao vocabulário da categoria "Experiência Profissional", uma palavra que correspondeu ao vocabulário "Educação" e por fim três palavras corresponderam ao vocabulário "Outros". Conclui-se então, usando soma dos valores *TF-IDF* das correspondências, que a linha foi bem classificada.

Observando os valores de *TF-IDF* para as correspondências feitas com a categoria "Informação Pessoal", nota-se que esses valores se encontram baixos relativamente ao que acontece com as correspondências feitas com outras categorias. Por essa razão, foi necessário observar quais as palavras categorizadas com "Informação Pessoal" (Figura 18).

```
(0, 1093)     0.2864252001675413 - 'sex female birth'
(0, 1092)     0.2864252001675413 - 'sex female'
(0, 1091)     0.2864252001675413 - 'sex'
(0, 950)      0.2243292118704479 - 'portuguese'
(0, 822)      0.2243292118704479 - 'nationality portuguese'
(0, 817)      0.1622332235733545 - 'nationality'
(0, 399)      0.2864252001675413 - 'female birth date'
(0, 398)      0.2864252001675413 - 'female birth'
(0, 397)      0.2864252001675413 - 'female'
(0, 247)      0.2864252001675413 - 'date nationality portuguese'
(0, 246)      0.2864252001675413 - 'date nationality'
(0, 240)      0.1622332235733545 - 'date'
(0, 102)      0.2864252001675413 - 'birth date nationality
(0, 101)      0.2864252001675413 - 'birth date'
(0, 100)      0.1622332235733545 - 'birth'
```

Figura 18 – Exemplo de palavras correspondidas com a categoria original

Após observação dos resultados constatou-se que todas essas palavras são palavras que caracterizam bem a categoria de "Informação Pessoal" e, por essa razão, estão presentes em quase todos os documentos relativos a essa categoria o que torna o seu valor de *TF-IDF* mais baixo.

De seguida, foi necessário observar quais as palavras com correspondências nas outras categorias para tentar perceber a razão dos seus valores serem mais altos do que na categoria classificada (Figura 19).

```
Categoria WE:
  (0, 2198)    0.7904489597281997 - 'portuguese'
  (0, 682)     0.6125279112535257 - 'date'

Categoria ED:
  (0, 946)     1.0 - 'portuguese'

Categoria OT:
  (0, 8645)    0.35566114128921 -'portuguese'
  (0, 2590)    0.6608725870305322 - 'date'
  (0, 1116)    0.6608725870305322 - 'birth'
```

Figura 19 – Exemplos de correspondências com outras categorias

Estas correspondências mostram que não são palavras que descrevem qualquer uma destas categorias e, por isso, são palavras que não estão geralmente presentes nas mesmas pelo que este método as classifica como sendo importantes. Em contrapartida linhas que pertencem a uma certa categoria e que são classificadas como tal irão ter mais correspondências, mas essas com valores *TF-IDF* mais baixos. Por outro lado, linhas que são classificadas como sendo de outra categoria que não a original vão apresentar palavras com valores de *TF-IDF* mais altos, no entanto, irão ter menos correspondências, fazendo com que a soma de todos valores classifique a linha corretamente.

Concluindo, estas falhas são obtidas devido ao facto de existirem palavras nos vocabulários das quatro categorias que são consideradas, pelo *TF-IDFVectorizer*, simultaneamente importantes para dois ou mais vocabulários, o que faz com que haja mais uma vez classificação errada de linhas (Figura 20).

```
Linha : 2003 –  Master of Science in Technology (graduated with distinction) - 2/23/2009
Categoria : ED

Correspondência PI :
Correspondência WE  :
        (0, 2561)    0.7904489597281997
        (0, 1732)    0.6125279112535257
Correspondência ED  :
        (0, 369)     1.0
Correspondência OT  :
        (0, 10702)   0.4924680993760029
        (0, 9938)    0.6147396913095939
        (0, 9906)    0.37019650744241195
        (0, 6857)    0.4924680993760029
```

Figura 20 – Exemplo de classificação errada de linha

Outra das razões que podem ter levado à falha desta abordagem prende-se também com o facto de, mais uma vez, os vocabulários terem um número de palavras não muito extenso devido ao tamanho do Corpus e dos CVs contidos nele. Para esta abordagem funcionar sem falhas os vocabulários teriam que ser constituídos por todas as combinações de palavras que os inúmeros candidatos possam utilizar para redigir os seus currículos, tarefa essa praticamente impossível. Para finalizar, os blocos de informação dividida estão a ser tratados pelo *TF-IDFVectorizer* como sendo apenas um documento por categoria, i.e., todos os blocos de "Informação Pessoal" dos vários CVs são compilados num só para a utilização dos dicionários, e por essa razão o peso das palavras pode estar a ser enviesado, pois este método calcula a importância de uma mesma palavra em vários documentos.

Não obstante esta abordagem foi a melhor das três já apresentadas até este ponto devido à implementação do método *TF-IDFVectorizer* e da sua metodologia de importância das palavras, neste caso por categoria.

4.4. Divisão de currículos por títulos

Em consequência de as abordagens anteriores apresentarem falhas aquando da divisão automática de um CV por linhas, tentou-se implementar um método que classifica blocos de categorias entre títulos.

Numa primeira fase desta abordagem foi relevante observar quais as várias maneiras em que os títulos das diversas categorias podem ser apresentados nos diversos CVs. Por exemplo, um candidato pode referir-se à categoria de "Informação Pessoal" de inúmeras maneiras tais como "Detalhes Pessoais", "Identificação", etc. Devido a esta observação tornou-se imperativo fazer um levantamento, entre todos os CVs do conjunto de treino do Corpus, de todos os possíveis títulos para cada categoria, tendo sido criada uma lista de todos os possíveis títulos por categoria.

Decidiu-se então implementar um código que recebe como *input* um CV, que é lido linha a linha. Se uma linha corresponder a um título da lista de títulos pertencente a essa categoria, essa linha é, com certeza, um título e, por essa razão, até ser lida uma nova linha que corresponda a um título de outra categoria todas as linhas, exceto a do novo título, vão ser categorizadas como sendo da categoria do primeiro título lido pelo código.

À primeira vista, esta abordagem devolveu bons resultados, no entanto isto deveu-se ao facto de os testes terem sido feitos apenas com um grupo reduzido de CVs do conjunto de treino do Corpus. Ao se testar com elementos fora desse conjunto, uma vez que alguns títulos descritos nestes novos CVs não estavam presentes na lista inicial de títulos, existem várias falhas.

Logo esta abordagem seria viável apenas se os títulos das várias categorias fossem imutáveis e fossem escritos do mesmo modo em todos os CVs, ou se a lista de títulos tivesse todos os títulos possíveis por categoria.

4.5. Divisão de currículos por títulos com recurso a expressões regulares

Após as falhas apresentadas na abordagem anterior foi necessário aprimorá-la para evitar essas falhas. Para isso foi criada uma nova abordagem, baseada em expressões regulares[3]. Utilizando as informações do levantamento feito aos títulos, na abordagem anterior, decidiu-se criar um código que para cada CV verifica qual o padrão de títulos utilizado pelo candidato. Por exemplo, se o candidato utiliza como primeiro título "Informação Pessoal" é muito provável que no resto do CV a escrita de títulos obedeça ao padrão inicial.

Com estas informações foram criados então padrões de escrita de títulos utilizando expressões regulares:

- Título escrito todo em letras maiúsculas (Ex. "INFORMAÇÃO PESSOAL).
- Título escrito todo em letras minúsculas (Ex. "informação pessoal).
- Título escrito com apenas a primeira letra da primeira palavra em maiúscula (Ex. "Informação pessoal").
- Título escrito com apenas as primeiras letras de cada palavra em maiúscula (Ex. "Informação Pessoal").
- Título escrito com padrão não contemplado (Padrão para salvaguardar erros por não ter sido encontrado um padrão válido).

[3] (https://docs.python.org/2/library/re.html).

Após a criação dos cinco padrões foi necessário usar outras expressões regulares para verificação do primeiro título sendo este o mais importante pois é com base neste que os seguintes vão ser encontrados. As expressões regulares implementadas para a ajuda na identificação do primeiro título, em conjunto com as dos padrões, foram então:

- Expressão que verifica se a linha contém mais que "X" palavras (Neste caso foi utilizado o valor de "X" como sendo 5 pois aquando do levantamento dos possíveis títulos nenhum deles apresentava mais do que cinco palavras).
- Expressão que verifica qual o caractere em que a linha acaba (Ex. Se acaba em ".","":"","", etc.)
- Expressão que verifica qual o espaçamento entre a margem e o início da linha ou possível título. (Ex. Os títulos têm sempre menos espaçamento do que qualquer outra linha num CV).

Para sofisticar ainda mais a captura, sem falhas, do padrão de títulos presente num CV, sem falhas, foi também adicionada a esta abordagem a anterior lista de possíveis títulos. Após todas estas refinações para aperfeiçoamento do código ficou então finalizada esta abordagem, que consiste na leitura de um CV linha a linha e com base em expressões regulares tenta capturar o padrão de títulos para que, quando um título é capturado, todas as linhas do CV sejam testadas com a respetiva expressão, ou expressões, fazendo com que o resto dos títulos dos CVs sejam capturados.

Examinando o CV da Figura 5 observamos que os títulos a encontrar são "Personal Information", "Professional Experience", "Education", "Competencies" e por fim "Comprehention", sendo que todos foram capturados pela nova abordagem (Figura 21).

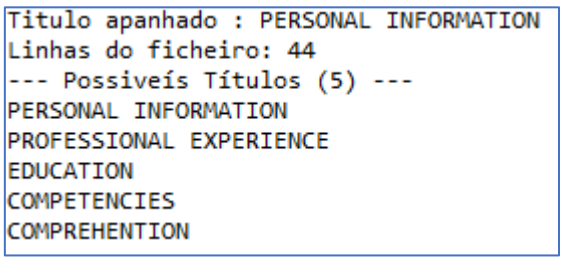


Figura 21 – Títulos capturados para o CV Exemplo através desta implementação

De acordo com a Figura 21 o primeiro título capturado pela implementação acima referida foi "Personal Information" escrito em maiúsculas, obedecendo ao primeiro

padrão referido. Por conseguinte, todas as linhas foram testadas para obedecer a este padrão e, as que obedeceram foram consideradas como títulos.

Contudo esta abordagem apresenta também ela lacunas que comprometem a correta divisão de um CV por categorias. Ao ser implementado um sistema de captura de padrão de títulos foi resolvido o problema da anterior abordagem. No entanto, mesmo sendo todos os títulos capturados, existem linhas que não são títulos e que, por obedecerem ao padrão são consideradas como tal (Figura 22).

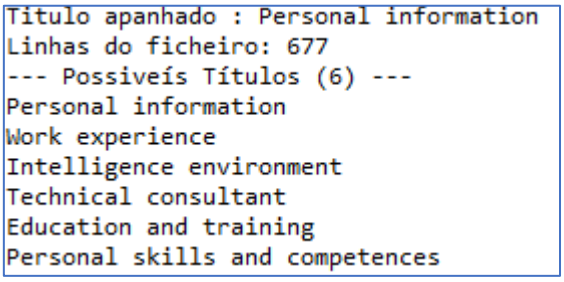


Figura 22 – Títulos capturados erradamente

Observa-se na Figura 22 que o título foi capturado com sucesso, tal como o seu padrão, no entanto existem linhas que não são títulos, mas que foram consideradas como sendo, devido à forma como o CV está redigido e formatado (Figura 23).

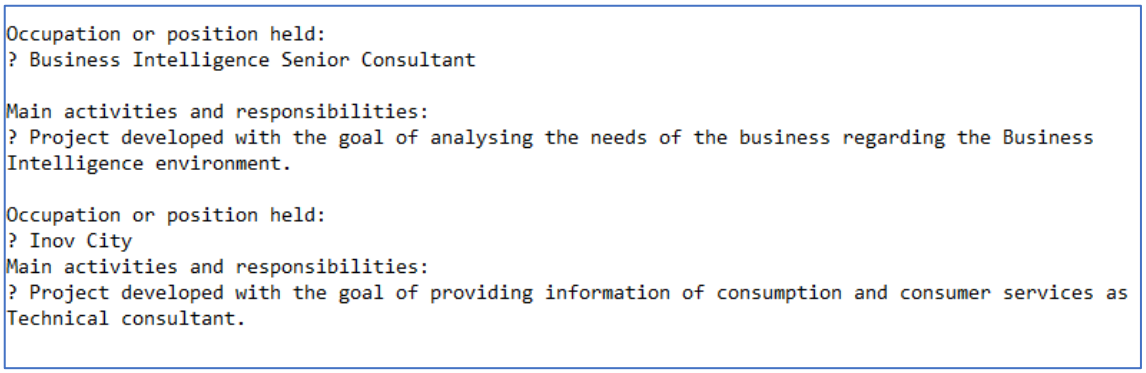


Figura 23 – Exemplo de título classificado erradamente

Por consequência, mesmo esta abordagem, sendo melhor que a anterior, é ainda facilmente falível.

4.6. Sumário

Foram apresentadas neste capítulo várias abordagens para dividir um CV pelas quatro categorias apresentadas em 3.1). Todas apresentaram resultados pouco satisfatórios devido a vários fatores que se prendem com cada uma delas, sendo o maior fator a simplicidade destas abordagens. Como é possível ver no ponto 4.1) esta abordagem falha simplesmente pelo facto de as mesmas palavras pertencerem a duas categorias, sendo que, para funcionar seria necessário existirem listas perfeitas para cada categoria, sem interseção de palavras nas diversas categorias. No ponto 4.2) tentou-se melhorar a abordagem anterior com recurso a dicionários. Notaram-se, de facto, melhorias mas, no entanto surge mais uma vez a questão da interseção de palavras nos dicionários das várias categorias, acabando por existir, ainda que em menor grau, o mesmo problema da abordagem anterior. Na abordagem 4.3) tentou-se novamente melhorar a abordagem antecedente, com recurso a uma ferramenta automática de cálculo da importância de uma palavra num documento. Mas, mais uma vez, esta abordagem apresenta falhas devido ao mesmo fator das que lhe antecedem, palavras que são simultaneamente classificadas como importantes para mais do que uma categoria. Devido a todas estas abordagens terem falhado em consequência da dependência das palavras contidas nas categorias, partiu-se para uma nova abordagem a este problema, a de tentar dividir um CV pelos títulos que contém. A primeira das duas abordagens relativas a este assunto – 4.4) – mostrou-se eficiente. No entanto quando apresentados títulos fora do conjunto pré-definido dos mesmos, esta abordagem falha redondamente. Por essa razão implementou-se uma última abordagem que não necessita de um conjunto pré-definido de títulos, mas ao invés disso, tenta capturar os títulos presentes num CV, para seguidamente dividir o CV com base neles, sendo esta a abordagem a melhor conseguida das cinco implementadas. No entanto, também esta abordagem é falível, devido ao fator ortográfico, i.e., simplesmente devido ao facto de como um CV está redigido.

Capítulo 5. Topic Modeling

Em virtude de as abordagens anteriores, baseadas em "*hand-crafted features*" e expressões regulares, apresentarem diversas lacunas e não serem práticas do ponto de vista de um ambiente real de uma empresa decidiu-se tratar da resolução desta questão com recurso a ferramentas de aprendizagem automática, devido à necessidade de automatização do processo (*Machine Learning*[4]).

Numa primeira fase para esta temática foi priorizada uma abordagem com recurso a *Topic Modeling*. Resumidamente, *Topic Modeling* é o processo de extração de tópicos presentes numa coleção de documentos com recurso a diversos métodos, para que seja mais fácil perceber quais os assuntos que neles estão contidos. Embora existam várias técnicas utilizadas para *Topic Modeling* (Quan, Liu, Lu, Ni, & Wenyin, 2010; Yih, Toutanova, Platt, & Meek, 2011), nesta abordagem iremos utilizar as três mais comuns para a realização desta tarefa, sendo elas *"Latent Dirichlet Allocation"* (LDA) (Blei, Ng, & Jordan, 2000), *"Latent Semantic Analysis"* (LSI) (Landauer, Folt, & Laham, 1998) e, por fim, *"Non-Negative Matrix Factorization"* (NMF) (Lee & Seung, 2001) (Baxter, Hastings, Law, & Glass, 2008).

Muito resumidamente:

- O modelo LDA é um modelo probabilístico utilizado para representar documentos através de um conjunto de tópicos, no qual cada tópico é representado com uma certa probabilidade.
- O modelo LSI é um modelo estatístico que examina dados num conjunto de documentos e o que neles está contido para gerar um conjunto de informações relativas a esses documentos e ao seu conteúdo.
- O modelo NMF é um modelo que gera um conjunto de tópicos para representar os documentos que lhe são dados como input.

Para todos estes modelos foram feitas duas implementações: uma com recurso a *TF-IDFVectorizer*, já referido em anteriores abordagens nesta dissertação, e outra com recurso a *CountVectorizer*[5].

[4] (https://www.sas.com/en_id/insights/analytics/machine-learning.html)

[5] (http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html)

O *CountVectorizer* é um método relativamente parecido com o *TF-IDFVectorizer*, no entanto, em vez de utilizar o valor *TF-IDF* para atribuir importância a uma palavra, utiliza a contagem de palavras num documento para esse fim, i.e., a importância de uma palavra está diretamente ligada à quantidade de vezes que aparece. Quanto mais vezes aparece, mais importante será.

Relativamente ao número de tópicos devolvidos pelos modelos utilizados, e para classificar uma categoria, o número mínimo de tópicos deve ser previamente definido. Para os testes, esse número mínimo foi de quatro, devido ao facto de ser o número de categorias existentes. Foram efetuados testes com 4, 5 e 6 tópicos, para observar se estes modelos identificariam nos documentos novas categorias. O número de palavras por tópico é também uma variável mutável, pelo que foram testados todos os modelos para seis, oito, dez e doze palavras por tópico. Como se irá mostrar, os melhores resultados foram obtidos com dez palavras por tópico, correspondendo a uma melhor descrição características para cada categoria.

5.1. *Topic Modeling* com recurso a modelo LDA

Usando *CountVectorizer*

O primeiro modelo a ser estudado foi o modelo LDA, com recurso ao método *CountVectorizer*. Para este modelo, foram efetuados testes com quatro, cinco e seis tópicos, sendo que o número de palavras mais adequado, tal como foi dito anteriormente, foi dez. Com menos palavras, uma descrição possível por categorias não era conseguida, e com mais de dez não foram observadas melhorias para esta descrição.

Para esta experiência, foram usados os CVs do conjunto de treino previamente divididos por categoria, assumindo que facilitaria a extração de tópicos correlacionados com as categorias existentes.

A primeira etapa de utilização deste modelo passa pelo estudo de qual o número de tópicos ideal para a identificação das categorias existentes. Utilizando quatro tópicos com dez palavras por tópico:

Utilizando quatro tópicos com dez palavras por tópico:

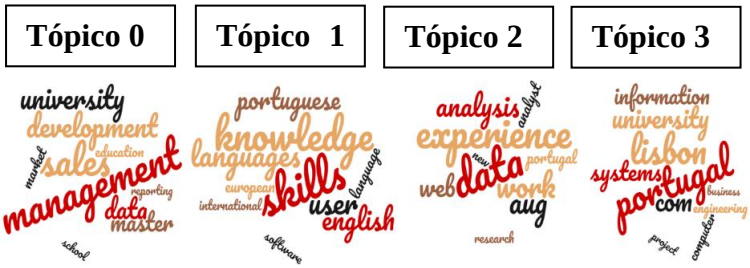


Figura 24 – Word clouds de tópicos – LDA CountVectorizer quatro tópicos

Utilizando cinco tópicos com dez palavras por tópico:

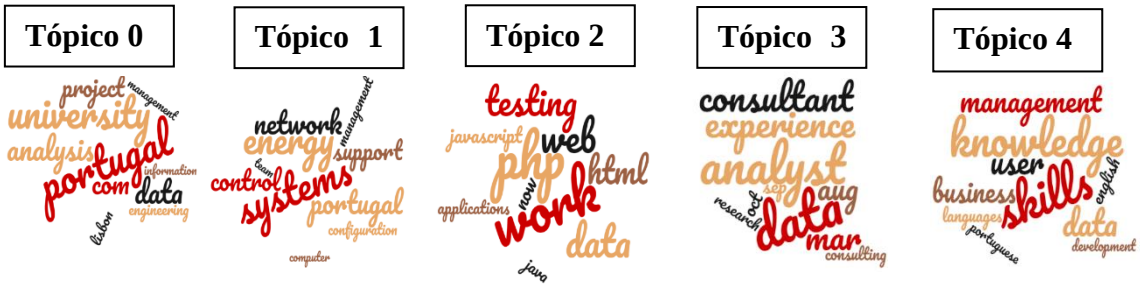


Figura 25 – Word clouds de tópicos – LDA CountVectorizer cinco tópicos

Utilizando seis tópicos com dez palavras por tópico:

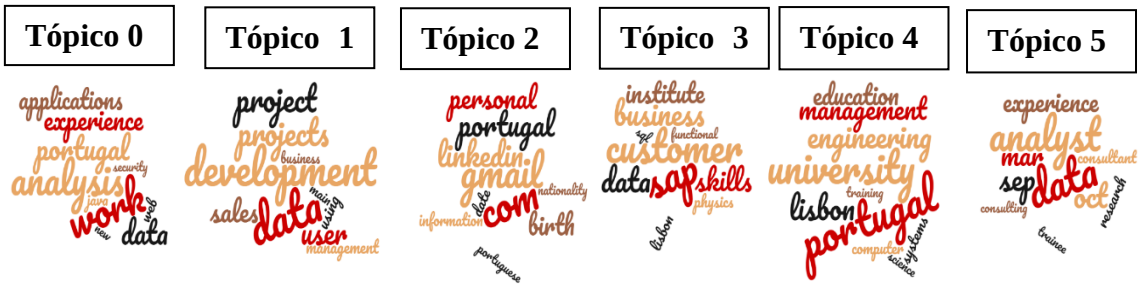


Figura 26 – Word clouds de tópicos – LDA CountVectorizer seis tópicos

Verifica-se que, à medida que o número de tópicos vai aumentando, vão variando as palavras presentes nos mesmos e também a qualidade da classificação. Para quatro tópicos (Figura 24), facilmente se relacionam as palavras presentes em cada tópico com as categorias de "Educação" – Tópico 3, "Outros" – Tópico 1, e "Experiência Profissional" – Tópico 2. No entanto, os tópicos contêm palavras de outras categorias, fazendo assim com que não seja claro que categoria cada tópico categoriza. Quando o número de tópicos é cinco (Figura 25), já apresentam uma descrição um pouco mais conseguida ou indicativa da possível categoria, com menos dispersão de palavras. Podemos fazer as seguintes associações por categoria – tópico: "Educação" – Tópico 0,

"Experiência Profissional" – Tópicos 1, 2 e 3, "Outros" – Tópico 4. No entanto, a categoria "Informação Pessoal" ainda não é percetível através deste método. Finalmente, ao serem utilizados seis tópicos para a categorização (Figura 26), vemos, então, presentes todas as categorias com um grau de detalhe muito maior que os anteriores, sendo que os Tópicos 0, 1, e 5 correspondem a "Experiência Profissional", o Tópico 2 a "Informação Pessoal", o Tópico 4 a "Educação" e por fim o Tópico 2 a "Outros".

Usando *TF-IDFVectorizer*

Foram efetuados novos testes de modelação de tópicos usando LDA com recurso ao método *TF-IDFVectorizer*. Foram, novamente, estudados os casos da variação de palavras com seis, oito, dez e doze palavras e, também para este modelo, os melhores resultados obtidos foram, novamente, com a utilização de dez palavras por tópico. No estudo do número ideal de tópicos para classificação de uma categoria utilizando esta abordagem foram obtidos os seguintes resultados:

Utilizando quatro tópicos com dez palavras por tópico:

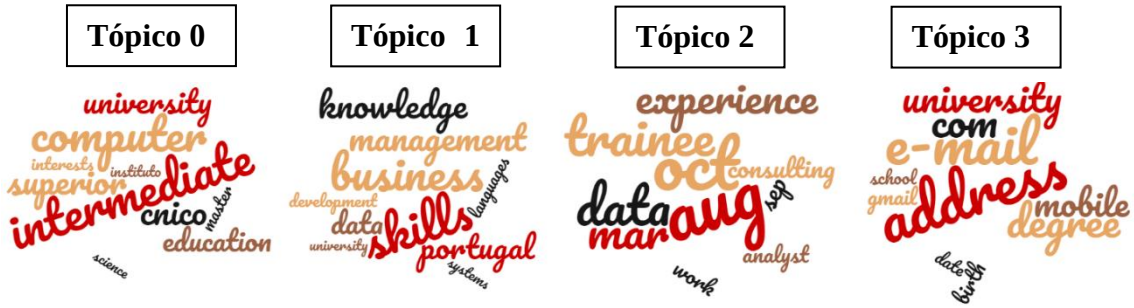


Figura 27 – Word clouds de tópicos – LDA TF-IDFVectorizer quatro tópicos

Utilizando cinco tópicos com dez palavras por tópico:

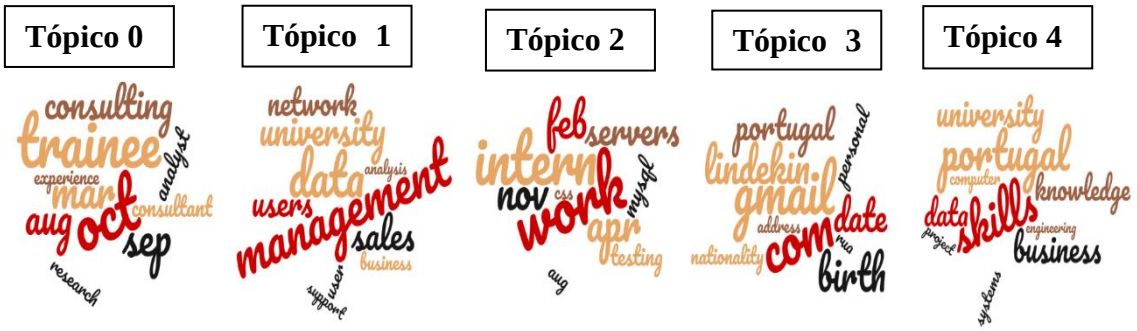


Figura 28 – Word clouds de tópicos – LDA TF IDFVectorizer cinco tópicos

Utilizando seis tópicos com dez palavras por tópico:

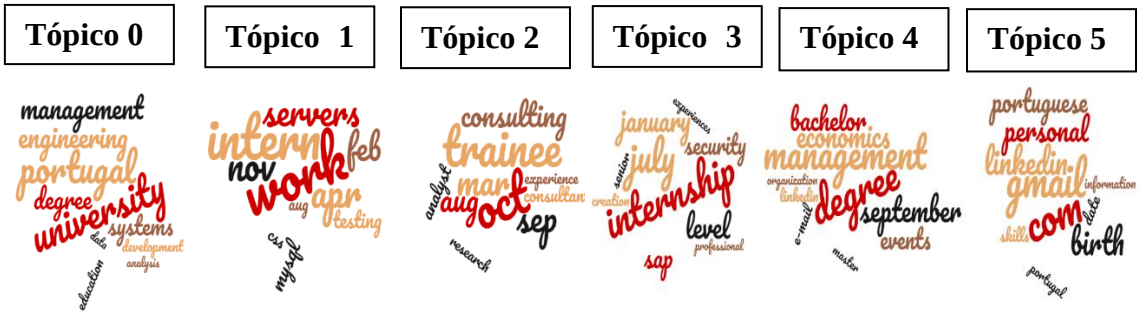


Figura 29 – Word clouds de tópicos – LDA TF-IDFVectorizer seis tópicos

Com o *TF-IDFVectorizer* a diferença é notória logo na primeira categorização utilizando quatro tópicos (Figura 27): as categorias presentes são bastantes explícitas e fáceis de identificar, tais como "Informação Pessoal" – Tópico 3, "Experiência Profissional" – Tópico 2 e "Outros" – Tópico 1. Para 5 tópicos (Figura 28) é possível observar novamente explícitas as categorias: "Informação Pessoal" – Tópico 3, "Experiência Profissional" – Tópicos 0 e 2 e "Outros" – Tópico 4. No entanto, existem categorias que não são reconhecidas, ou não estão presentes, tais como a categoria "Educação". Só ao utilizar seis tópicos para classificação é que esta categoria surge bem detalhada, ao mesmo nível de detalhe das outras (Figura 29). Neste caso, todas as categorias são facilmente identificáveis: "Informação Pessoal" – Tópico 5, "Experiência Profissional" – Tópicos 1 e 2, "Educação" – Tópico 0 e 4, "Outros" – Tópico 3 e 4.

5.2. *Topic Modeling* com recurso a modelo LSI

Usando *CountVectorizer*.

O segundo modelo em estudo para esta abordagem foi o modelo LSI com recurso a *CountVectorizer*. Mais uma vez foi estudado qual o melhor número de palavras para categorização dos tópicos com o mesmo número de palavras que o anterior modelo: seis, oito, dez e doze palavras.

Utilizando quatro tópicos com dez palavras por tópico:

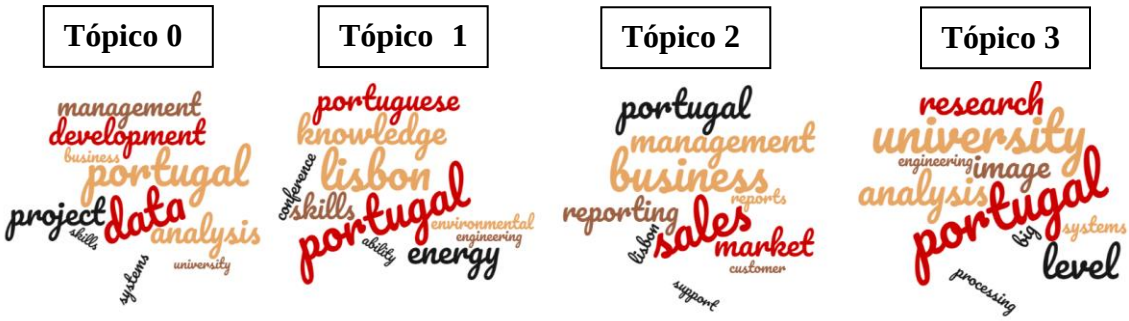


Figura 30 – Word clouds de tópicos – LSI CountVectorizer quatro tópicos

Neste modelo quando é aumentado o número de tópicos apenas é criado um novo tópico, i.e., para cinco tópicos os quatro primeiros são iguais aos criados aquando da utilização de quatro tópicos, e assim sucessivamente.

Utilizando cinco tópicos com dez palavras por tópico:

Figura 31 – Word cloud de novo tópico – LSI CountVectorizer cinco tópicos

Utilizando seis tópicos com dez palavras por tópico:

Figura 32 – Word cloud de novo tópico – LSI CountVectorizer seis tópicos

Para este modelo, e como podemos observar pelas figuras (Figura 30 a 32) as associações possíveis foram praticamente inexistentes pois, para qualquer combinação entre número de tópicos e palavras utilizadas, o modelo gera tópicos com palavras

pertinentes às várias categorias, não sendo possível classificar os tópicos quanto à categoria a que se referem.

Usando *TF-IDFVectorizer*

O processo anterior foi repetido mais uma vez, mas neste caso com recurso ao método *TF-IDFVectorizer*.

Utilizando quatro tópicos com dez palavras por tópico:

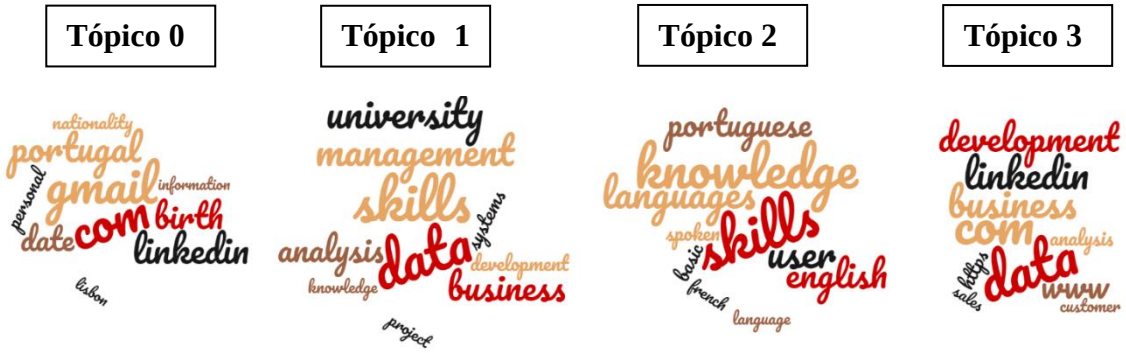


Figura 33 – Word clouds de tópicos – LSI TF-IDFVectorizer quatro tópicos

Mais uma vez, aquando da incrementação do número de tópicos neste modelo, apenas o novo tópico criado difere dos anteriores.

Utilizando cinco tópicos com dez palavras por tópico:

Figura 34 – Word cloud de novo tópico – LSI TF-IDFVectorizer cinco tópicos

Utilizando seis tópicos com dez palavras por tópico:

Figura 35 – Word cloud de novo tópico – LSI TF-IDFVectorizer seis tópicos

Com a implementação do *TF-IDFVectorizer* para este modelo podemos observar que houve uma melhoria aquando da criação dos tópicos (Figura 33 a Figura 35). No entanto, essa melhoria reflete-se apenas na identificação de uma categoria, "Informação Pessoal" – Tópico 3. Para todas as restantes categorias não é possível a sua identificação nos tópicos criados.

5.3. *Topic Modeling* com recurso a modelo NMF

Usando *CountVectorizer*

Mais uma vez o processo que foi realizado para os anteriores modelos foi também ele repetido na implementação deste, primeiramente utilizando *CountVectorizer*.

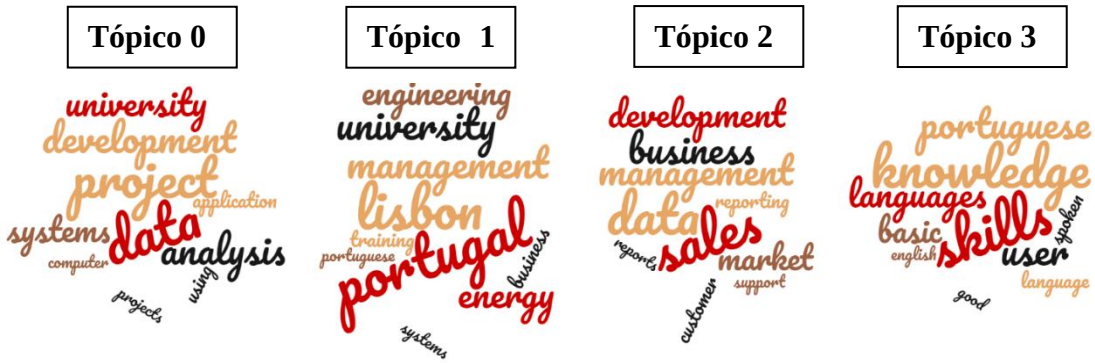


Figura 36 – Word clouds de tópicos – NMF CountVectorizer quatro tópicos

Também para este modelo quando é incrementado o número de tópicos é criado um novo tópico, diferente dos anteriores, no entanto os já existentes voltam a repetir-se.

Semelhantemente ao que já tinha sido observado com o método *CountVectorizer* para outros modelos, o resultado para qualquer número de tópicos independentemente do número de palavras utilizadas não é conclusivo aquando da classificação de uma categoria através dos mesmos (Figura 36).

Usando *TF-IDFVectorizer*

Por último foi testado o modelo NMF utilizando o método *TF-IDFVectorizer*. De novo testado nas mesmas circunstâncias que todos os outros modelos.

Utilizando quatro tópicos com dez palavras por tópico:

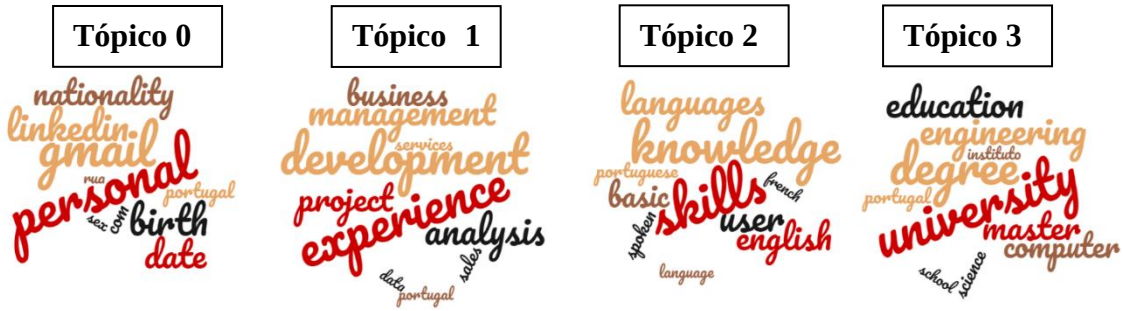


Figura 37 – Word clouds de tópicos – NMF TF-IDFVectorizer quatro tópicos

Considerando os resultados na Figura 37 podemos, então, inferir que este modelo, com o uso do método *TF-IDFVectorizer*, apresenta muito melhores resultados que os obtidos para o mesmo modelo através do método *CountVectorizer*. Logo na primeira criação de tópicos (utilizando apenas quatro grupos), são facilmente identificáveis as quatro categorias uma por tópico, sem confusão ou sobreposição: "Informação Pessoal" – Tópico 0, "Experiência Profissional" – Tópico 1, "Educação" – Tópico 3, "Outros" – Tópico 2. O aumento do número de tópicos para cinco ou seis, nesta implementação, não contribui para o melhoramento da mesma e apenas faz com que sejam criados tópicos diferentes, mas associáveis a categorias repetidas.

5.4. Classificação de linhas com recurso a *Topic Modeling*

Após análise das conclusões obtidas nos pontos 5.1) a 5.3) desta dissertação, para a classificação de linhas do CV segundo as categorias a que pertencem, foram considerados os resultados obtidos com o *Topic Modeling*. A decisão recaiu nos modelos NMF com

TF-IDFVectorizer para quatro tópicos e no modelo LDA com *TF-IDFVectorizer* para seis tópicos. Estes foram, sem dúvida, os melhores resultados obtidos quando da utilização de *Topic Modeling* para classificação de categorias.

Para a classificação por linhas, nesta abordagem foi implementado código para que, lendo um CV linha a linha, seja utilizada a informação que foi previamente obtida através da criação dos tópicos pelos modelos para categorizar uma linha.

Este código recebe então como input o modelo que vai ser utilizado, o conjunto de dados pré-treinados, o número de tópicos que vão ser utilizados para a classificação e, por fim, o CV no qual as linhas vão ser classificadas, devolvendo como output o CV, em formato HTML, com as respetivas linhas classificadas por categoria de acordo com o modelo utilizado.


Curriculum Vitae Jane Doe
PERSONAL INFORMATION
Av. da Liberdade, 1250-096 Lisboa
+351 912 345 678
jane.doe@gmail.com
Sex Female | Birth date 01/01/1985 | Nationality Portuguese

PROFESSIONAL EXPERIENCE

Date (2013 - Today) Consultant

SAP Portugal – Near Shore Center Lisboa

Certifications and Trainings:

? Certified consultant in SAP FI ERP 6.0
? Installation and Migration Bootcamp to Simple Finance 1503
? SEPA Certified
? IUT 110 Business Processes in SAP for Utilities
? IUT 210 Master Data and Basic Functions
? IUT 220 Device Management
? IUT 230 Billing and Invoicing
? IUT 240 FICA

Dates (2009 - 2009) - Promoter AA.Com

? Participation and team management in promotion actions during August for the launch of the newspaper "i".
EDUCATION

Dates (2004 - 2009) Finance degree

ISEG Instituto Superior de Economia e Gestão

COMPETENCIES
Mother Language Portuguese

Other languages*
COMPREHENTION
Oral Comprehention Reading Oral interaction Oral reproduction

Inglês C2 C2 C1 C1
Espanhol B1 B1 B2 B1
Francês A2 A2 A2 A1

Figura 38 – Exemplo de CV dado como output utilizando o modelo LDA

A razão pela qual o CV é dado pelo output em formato HTML deve-se ao facto de ter ser sido adicionada ao código uma função que vai colorir as linhas de acordo com a sua categoria para melhor perceção dos conjuntos que estão presentes num CV, i.e., se linhas da mesma cor se encontram agrupadas significa que a classificação foi feita corretamente. As cores são atribuídas aleatoriamente e podem variar de teste para teste, ou seja, num teste para um CV, a "Informação Pessoal" pode ser colorida de verde e no seguinte esta pode ficar colorida de vermelho. O número máximo de cores que podem coexistir num output corresponde ao número máximo de tópicos, sendo que se existem dois tópicos relativos à mesma categoria, estes serão pintados com cores diferentes.

Primeiramente foi testado o modelo LDA utilizando 6 tópicos (Figura 38) e, de seguida, testado para o modelo NMF utilizando quatro tópicos (Figura 39).


Curriculum Vitae Jane Doe
PERSONAL INFORMATION
Av. da Liberdade, 1250-096 Lisboa
+351 912 345 678
jane.doe@gmail.com
Sex Female | Birth date 01/01/1985 | Nationality Portuguese

PROFESSIONAL EXPERIENCE

Date (2013 - Today) Consultant

SAP Portugal – Near Shore Center Lisboa

Certifications and Trainings:

? Certified consultant in SAP FI ERP 6.0
? Installation and Migration Bootcamp to Simple Finance 1503
? SEPA Certified
? IUT 110 Business Processes in SAP for Utilities
? IUT 210 Master Data and Basic Functions
? IUT 220 Device Management
? IUT 230 Billing and Invoicing
? IUT 240 FICA

Dates (2009 - 2009) - Promoter AA.Com

? Participation and team management in promotion actions during August for the launch of the newspaper "i".
EDUCATION

Dates (2004 - 2009) Finance degree

ISEG Instituto Superior de Economia e Gestão

COMPETENCIES
Mother Language Portuguese

Other languages*
COMPREHENTION
Oral Comprehention Reading Oral interaction Oral reproduction

Inglês C2 C2 C1 C1
Espanhol B1 B1 B2 B1
Francês A2 A2 A2 A1

Figura 39 – Exemplo de CV dado como output utilizando o modelo NMF

Como é possível observar na figura referente à classificação de linhas utilizando o modelo NMF (Figura 39), as linhas da mesma cor permanecem mais agrupadas do que na figura referente ao modelo LDA (Figura 38), querendo com isto dizer que o melhor classificador de linhas foi obtido utilizando o modelo NMF com apenas quatro tópicos e com recurso a *TF-IDFVectorizer*.

Ainda assim, visto que a análise de *Accuracy* de um modelo não pode ser feita apenas pela observação dos CVs e do agrupamento de cores nos mesmos, foi necessário medir a *Accuracy* de todos os modelos quanto à sua correta classificação de linhas (Tabela 10). Esta *Accuracy* foi calculada pelo quociente do número de linhas classificadas corretamente pelo número total de linhas classificadas.

Tabela 10 – Análise de Accuracy dos diferentes modelos

	Accuracy
LDA c/ *TF-IDF*Vec – 6 Tópicos	53.07%
LDA c/ CountVec – 6 Tópicos	43.27%
LSI c/ *TF-IDF*Vec – 6 Tópicos	12.03%
LSI c/ CountVec – 6 Tópicos	24.51%
NMF c/ *TF-IDF*Vec – 4 Tópicos	61.23%
NMF c/ CountVec – 6 Tópicos	15.76%

Como seria de esperar, apenas o modelo NMF com 4 tópicos apresenta resultados positivos. De facto, este foi um dos que obteve a melhor categorização por tópicos. O modelo LDA com *TF-IDFVectorizer* é pouco superior a uma classificação aleatória.

5.5. Sumário

Como foi dito no início do corrente capítulo, estas abordagens surgiram devido à necessidade de implementação de ferramentas de aprendizagem automática para automatização e tentativa de melhoramento do processo de divisão de CVs.

Para isso foram criadas novas abordagens com recurso a *Topic Modeling*. Todos estes modelos foram testados com recurso à variabilidade de tópicos, com o intuito de concluir qual o melhor modelo para a resolução do problema da divisão de um CV por categorias. Após a conclusão de qual o número de tópicos ideal para cada modelo, foi implementada uma ferramenta para classificação de linhas de um CV. Com recurso a essa ferramenta,

foi possível observar qual a *Accuracy* de cada modelo aquando da classificação correta de linhas, sendo que os melhores resultados foram obtidos com o modelo *NMF* com recurso a *TF-IDFVectorizer*, obtendo uma *Accuracy* de 61.23%. Os valores relativos aos diferentes testes (Tabela 10) demonstram que, mesmo ao utilizar os melhores modelos para associação tópicos-categoria, a classificação do total de linhas nunca ultrapassa os 61.23%, o que ainda representa uma probabilidade pouco satisfatória.

Capítulo 6. Classificação Automática de CVs

Visto que todas as abordagens até ao presente capítulo apresentam falhas significativas na classificação de linhas de texto que compõem um CV, contribuindo para a não implementação de um classificador automático de CVs através das mesmas, decidiu-se enveredar por abordagens referentes à classificação automática de um CV, sem recurso às linhas do mesmo.

Para uma abordagem completamente automática, aquando da classificação de um CV, tendo em conta o seu conteúdo, foi necessário recorrer à utilização de várias medidas de similaridade entre documentos. As medidas escolhidas para esta abordagem foram então *"Cosine Similarity"*, *"Euclidean Distance"* e *"Jaccard Similarity Coefficient"*, que correspondem a medidas frequentemente utilizadas em tarefas de cálculo de similaridade entre documentos de texto (H.Gomaa & A. Fahmy, 2013; Huang, 2008; Mihalcea, Corley, & Strapparava, 2006; Suphakit Niwattanakul*, Jatsada Singthongchai, 2013).

A *"Cosine Similarity"*[6], ou semelhança do cosseno, é uma medida que calcula o valor do cosseno entre dois vetores. Para aplicação desta medida, os documentos são representados como vetores no espaço vetorial e o cosseno entre eles é calculado, sendo que quanto mais perto de 1 esta medida se encontra mais similares são os documentos.

A *"Euclidean Distance"*[7], ou distância Euclidiana, é a distância entre dois pontos presentes no espaço Euclidiano, devido a esse fator, os documentos representados por vetores necessitam de ser normalizados por meio da norma Euclidiana, para que possam ser representados como pontos no espaço Euclidiano, e, por conseguinte, a sua distância Euclidiana possa ser obtida, sendo que quanto maior é esta distância maior a diferença entre dois documentos.

O *"Jaccard Similarity Coefficient"*[8] é uma medida que calcula a similaridade entre, neste caso, documentos com base na divisão entre a interseção dos dados que dois documentos têm em comum e a união desses dois documentos, sendo que quanto mais alta é esta medida maior a similaridade entre os documentos em causa.

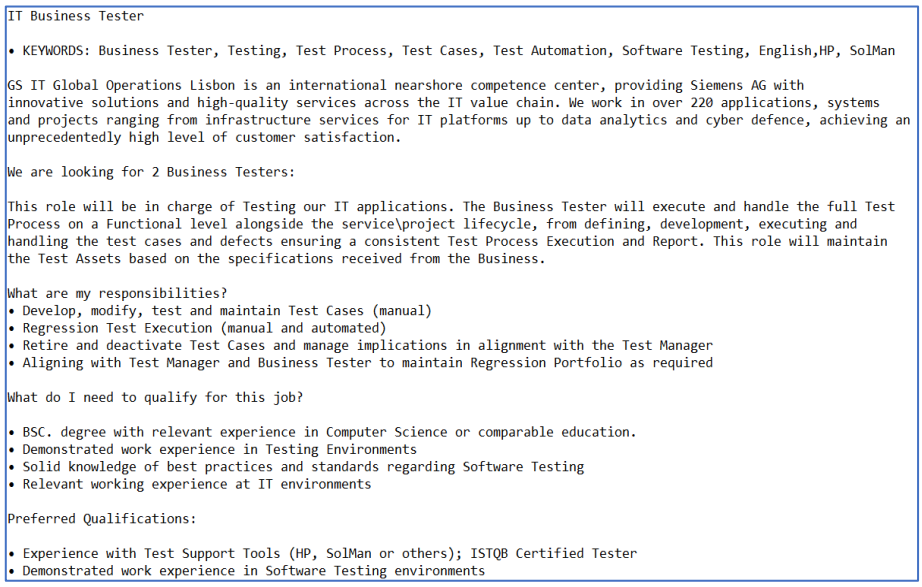
[6] (http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/)

[7] (https://hlab.stanford.edu/brian/euclidean_distance_in.html)

[8] (https://www.statisticshowto.datasciencecentral.com/jaccard-index/)

6.1. Classificação automática utilizando Job ID's

Para esta abordagem foi criado um classificador automático de CVs, que calcula, através das métricas acima descritas, a similaridade entre um determinado CV e a Job ID a que se encontra associado. A Figura 40 apresenta o exemplo de uma Job ID, que corresponde a um anúncio de trabalho, tal como previamente descrito no ponto 3.1).



IT Business Tester

• KEYWORDS: Business Tester, Testing, Test Process, Test Cases, Test Automation, Software Testing, English,HP, SolMan

GS IT Global Operations Lisbon is an international nearshore competence center, providing Siemens AG with innovative solutions and high-quality services across the IT value chain. We work in over 220 applications, systems and projects ranging from infrastructure services for IT platforms up to data analytics and cyber defence, achieving an unprecedentedly high level of customer satisfaction.

We are looking for 2 Business Testers:

This role will be in charge of Testing our IT applications. The Business Tester will execute and handle the full Test Process on a Functional level alongside the service\project lifecycle, from defining, development, executing and handling the test cases and defects ensuring a consistent Test Process Execution and Report. This role will maintain the Test Assets based on the specifications received from the Business.

What are my responsibilities?
• Develop, modify, test and maintain Test Cases (manual)
• Regression Test Execution (manual and automated)
• Retire and deactivate Test Cases and manage implications in alignment with the Test Manager
• Aligning with Test Manager and Business Tester to maintain Regression Portfolio as required

What do I need to qualify for this job?

• BSC. degree with relevant experience in Computer Science or comparable education.
• Demonstrated work experience in Testing Environments
• Solid knowledge of best practices and standards regarding Software Testing
• Relevant working experience at IT environments

Preferred Qualifications:

• Experience with Test Support Tools (HP, SolMan or others); ISTQB Certified Tester
• Demonstrated work experience in Software Testing environments

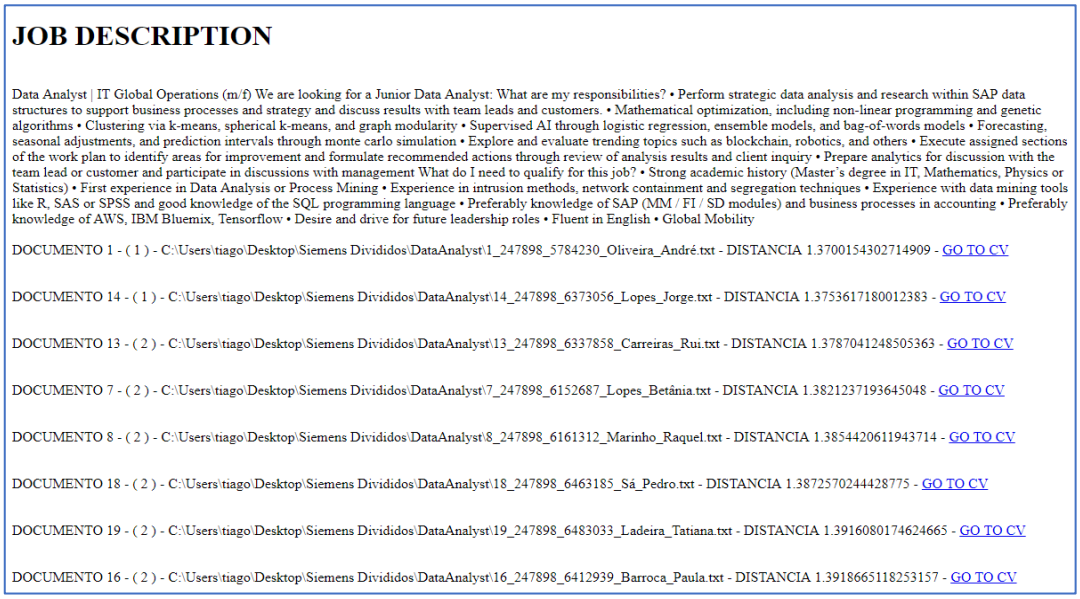
Figura 40 – Exemplo de Job ID

Devido ao facto de duas destas medidas utilizarem vetores no cálculo da sua similaridade, *"Cosine Similarity"* e *"Euclidean Distance"*, os CVs dados como input para essas medidas necessitam de ser apresentados na forma de vetor pelo que se optou por implementar os métodos de *CountVectorizer* e *TF-IDFVectorizer* para as duas medidas de modo a estudar qual o método com melhor desempenho. Para a terceira medida, *"Jaccard Similarity Coefficient"*, como não foram utilizados vetores para o seu cálculo, nenhum destes métodos foi implementado.

Para esta abordagem foram utilizados 20 CVs por departamento, sendo 20 departamentos diferentes, foram utilizados 400 CVs e 20 Job IDs.

Foi criada uma ferramenta para calcular a similaridade de todos os CVs de um departamento relativamente à sua respetiva Job ID com o intuito de perceber qual ou quais os candidatos que mais se adequam ao cargo referente à Job ID. Esta ferramenta recebe como input todos os CVs de candidatos a um cargo de um certo departamento, a Job ID desse departamento e ainda qual a métrica que irá ser utilizada para calcular a

similaridade. Posteriormente é devolvido como output um ficheiro HTML contendo a descrição da JobID, uma lista ordenada, do candidato mais adequado para o menos, e a respetiva distância à Job ID (Figura 41). Neste ficheiro é ainda possível visualizar os CVs dos candidatos, também eles ordenados por adequação, através de um link inserido à frente da lista.



JOB DESCRIPTION

Data Analyst | IT Global Operations (m/f) We are looking for a Junior Data Analyst: What are my responsibilities? • Perform strategic data analysis and research within SAP data structures to support business processes and strategy and discuss results with team leads and customers. • Mathematical optimization, including non-linear programming and genetic algorithms • Clustering via k-means, spherical k-means, and graph modularity • Supervised AI through logistic regression, ensemble models, and bag-of-words models • Forecasting, seasonal adjustments, and prediction intervals through monte carlo simulation • Explore and evaluate trending topics such as blockchain, robotics, and others • Execute assigned sections of the work plan to identify areas for improvement and formulate recommended actions through review of analysis results and client inquiry • Prepare analytics for discussion with the team lead or customer and participate in discussions with management What do I need to qualify for this job? • Strong academic history (Master's degree in IT, Mathematics, Physics or Statistics) • First experience in Data Analysis or Process Mining • Experience in intrusion methods, network containment and segregation techniques • Experience with data mining tools like R, SAS or SPSS and good knowledge of the SQL programming language • Preferably knowledge of SAP (MM / FI / SD modules) and business processes in accounting • Preferably knowledge of AWS, IBM Bluemix, Tensorflow • Desire and drive for future leadership roles • Fluent in English • Global Mobility

DOCUMENTO 1 - (1) - C:\Users\tiago\Desktop\Siemens Divididos\DataAnalyst\1_247898_5784230_Oliveira_André.txt - DISTANCIA 1.3700154302714909 - GO TO CV

DOCUMENTO 14 - (1) - C:\Users\tiago\Desktop\Siemens Divididos\DataAnalyst\14_247898_6373056_Lopes_Jorge.txt - DISTANCIA 1.3753617180012383 - GO TO CV

DOCUMENTO 13 - (2) - C:\Users\tiago\Desktop\Siemens Divididos\DataAnalyst\13_247898_6337858_Carreiras_Rui.txt - DISTANCIA 1.3787041248505363 - GO TO CV

DOCUMENTO 7 - (2) - C:\Users\tiago\Desktop\Siemens Divididos\DataAnalyst\7_247898_6152687_Lopes_Betânia.txt - DISTANCIA 1.3821237193645048 - GO TO CV

DOCUMENTO 8 - (2) - C:\Users\tiago\Desktop\Siemens Divididos\DataAnalyst\8_247898_6161312_Marinho_Raquel.txt - DISTANCIA 1.3854420611943714 - GO TO CV

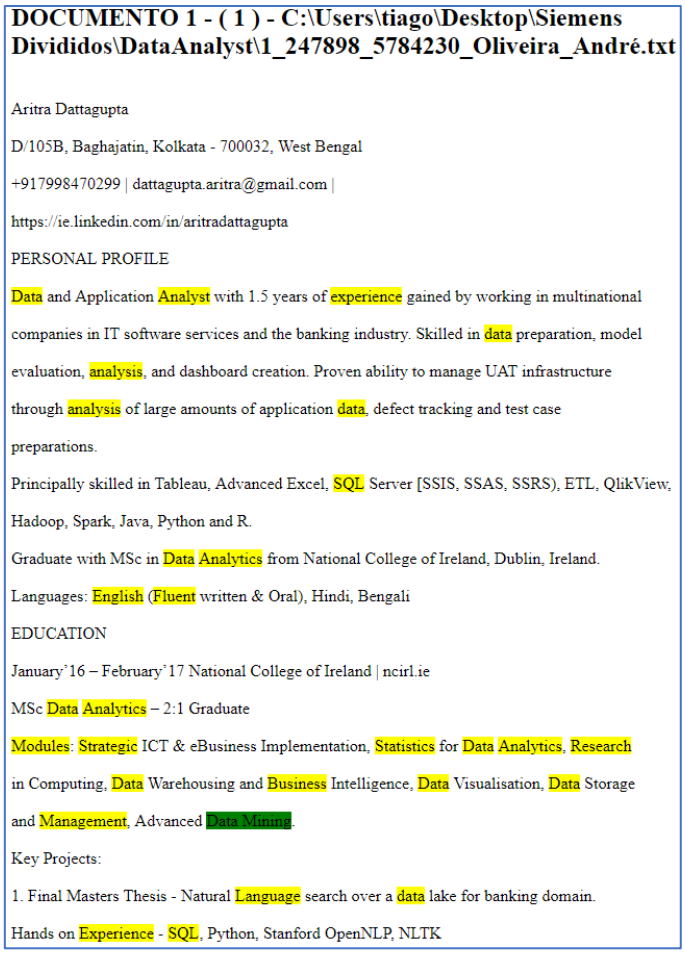
DOCUMENTO 18 - (2) - C:\Users\tiago\Desktop\Siemens Divididos\DataAnalyst\18_247898_6463185_Sá_Pedro.txt - DISTANCIA 1.3872570244428775 - GO TO CV

DOCUMENTO 19 - (2) - C:\Users\tiago\Desktop\Siemens Divididos\DataAnalyst\19_247898_6483033_Ladeira_Tatiana.txt - DISTANCIA 1.3916080174624665 - GO TO CV

DOCUMENTO 16 - (2) - C:\Users\tiago\Desktop\Siemens Divididos\DataAnalyst\16_247898_6412939_Barroca_Paula.txt - DISTANCIA 1.3918665118253157 - GO TO CV

Figura 41 – Exemplo de output

Devido ao facto destas métricas terem como base a similaridade de documentos, e por consequência, a similaridade entre palavras nos mesmos, foi adaptada a esta implementação a coloração das palavras que contribuem para uma maior similaridade aquando da visualização dos CVs no ficheiro dado no output (Figura 42).



DOCUMENTO 1 - (1) - C:\Users\tiago\Desktop\Siemens Divididos\DataAnalyst\1_247898_5784230_Oliveira_André.txt

Aritra Dattagupta

D/105B, Baghajatin, Kolkata - 700032, West Bengal

+917998470299 | dattagupta.aritra@gmail.com |

https://ie.linkedin.com/in/aritradattagupta

PERSONAL PROFILE

Data and Application Analyst with 1.5 years of experience gained by working in multinational companies in IT software services and the banking industry. Skilled in data preparation, model evaluation, analysis, and dashboard creation. Proven ability to manage UAT infrastructure through analysis of large amounts of application data, defect tracking and test case preparations.

Principally skilled in Tableau, Advanced Excel, SQL Server [SSIS, SSAS, SSRS), ETL, QlikView, Hadoop, Spark, Java, Python and R.

Graduate with MSc in Data Analytics from National College of Ireland, Dublin, Ireland.

Languages: English (Fluent written & Oral), Hindi, Bengali

EDUCATION

January'16 – February'17 National College of Ireland | ncirl.ie

MSc Data Analytics – 2:1 Graduate

Modules: Strategic ICT & eBusiness Implementation, Statistics for Data Analytics, Research in Computing, Data Warehousing and Business Intelligence, Data Visualisation, Data Storage and Management, Advanced Data Mining.

Key Projects:

1. Final Masters Thesis - Natural Language search over a data lake for banking domain.

Hands on Experience - SQL, Python, Stanford OpenNLP, NLTK

Figura 42 – Exemplo de visualização de CV através do output

Após a criação da ferramenta e da sua execução para todas as métricas foi necessário calcular qual o desempenho das mesmas, pois os CVs já se encontram classificados automaticamente, no entanto é desconhecido se estes estão a ser bem ou mal classificados, para isso, foram utilizadas respostas de recrutadores reais dos vários departamentos da Siemens, aos CVs em causa como meio de *cross-validation*.

Para a criação desta medida de *Accuracy* foram utilizadas as respostas dos recrutadores e as respostas dadas pela ferramenta criada, sendo que a *Accuracy* é obtida pelo quociente da quantidade de respostas que coincidem entre as duas partes, sobre a quantidade de respostas que não coincidem, i.e., se um recrutador de uma determinada Job ID avaliou cinco CVs como sendo bons, dez como medianos e cinco como sendo maus, então a ferramenta vai classificar os cinco mais similares como sendo bons, os dez seguintes como sendo medianos e os cinco últimos como sendo maus, mesmo que os cinco classificados automaticamente como bons difiram dos cinco classificados pelos recrutadores, e assim sucessivamente. Por esta razão foi criada uma lista numérica com

essas respostas, sendo que esta lista contém o número de "zeros" correspondentes ao número de maus candidatos, "uns" correspondentes ao número de bons candidatos e por fim "dois" correspondentes ao número de candidatos "medianos". Essa lista foi então adicionada ao input da ferramenta anteriormente criada para aquando do cálculo da similaridade, seja conhecido o número de CVs que têm de ser classificados como bons, maus e medianos. A ferramenta foi então adaptada para devolver uma matriz de confusão com três colunas e três linhas, sendo que as linhas representam as respostas dos recrutadores quanto aos CVs, e as colunas as respostas por parte da classificação automática dos CVs. A soma de todos os elementos da primeira linha da matriz correspondente ao número de CVs classificados como maus pelo recrutador, da segunda linha como bons e da terceira como medianos, seguindo a mesma ordem de ideias, mas desta vez para as colunas, a soma de todos os elementos da primeira coluna corresponde ao número de CVs classificados automaticamente como maus, da segunda coluna como bons e da última coluna como medianos. Assim, quando observamos a diagonal principal desta matriz temos então o número de respostas em que ambas as partes coincidem, i.e., o número de vezes que as respostas dadas pelos recrutadores são iguais às dadas pela classificação automática e com isso obter a *Accuracy* de cada métrica de similaridade (Figura 43). No caso concreto da matriz representada na figura, seria obtida uma *Accuracy* de 45%.

		Resposta das métricas		
		Maus	Bons	Medianos
Resposta dos Recrutadores	Maus	7	2	3
	Bons	3	1	0
	Medianos	2	1	1

Figura 43 – Exemplo de matriz de confusão devolvida pela ferramenta

Com recurso à ferramenta criada foi ainda gerada uma matriz de confusão que representa todas as classificações de uma determinada métrica para todas as Job IDs, i.e., existem vinte Job IDs, logo, existem vinte matrizes. Para ser mais clara a visualização das matrizes foi feita uma soma das mesmas dividindo pelo número de Job IDs obtendo assim

uma única matriz de confusão, denominada para este caso como matriz de confusão média.

Analisando a Tabela 11 podemos então ver que a classificação automática dos CVs foi pouco precisa, sendo que todas as métricas obtiveram resultados muito similares, no entanto o melhor resultado foi obtido através de *"Cosine Similarity"* utilizando o método *TF-IDFVectorizer*, querendo com isto dizer que a melhor classificação automática de um CV por via desta métrica tem uma probabilidade de 50.23% de estar correta. A razão deste baixo desempenho pode ser explicado pelo facto de estas métricas não conseguirem identificar o que é um bom CV e o que é necessário um CV conter para ser classificado como bom, sendo que a sua regra para classificarem os mesmos é baseada em palavras que estão presentes em ambos os documentos comparados, o que explica também o facto de o método com melhores resultados ser *"Cosine Similarity"*.

No entanto, para as métricas que lidam com vetores, *"Cosine Similarity"* e *"Euclidean Distance"*, estes testes foram feitos utilizando primeiramente um conjunto de "*n-grams*" entre um e três, ou seja, unigramas, bigramas e trigramas, e por esse motivo foi necessário testar todos estes modelos para contemplar a utilização isoladamente de unigramas, bigramas, trigramas, unigramas e bigramas, e por fim bigramas e trigramas (Tabela 11).

Tabela 11 – Análise da Accuracy das diferentes métricas utilizando n-gramas

	Euclidean Distance c/ *TF-IDF*Vec	*Euclidean Distance* c/ CountVec	*Cosine Similarity* c/ *TF-IDF*Vec	*Cosine Similarity* c/ CountVec	*Jaccard Similarity*
Unigramas	49.44%	46.43%	49.41%	49.54%	45.52%
Bigramas	43.76%	42.05%	42.37%	43.74%	
Trigramas	51.70%	45.19%	54.76%	54.26%	
Uni + Bigramas	49.52%	46.04%	49.09%	48.38%	
Bi + Trigramas	44.01%	42.05%	43.84%	43.39%	
Uni + Bi + Trigramas	49.52%	46.04%	50.23%	47.68%	

Com a implementação dos vários conjuntos de n-grams podemos observar pela Tabela 11 que a *Accuracy* melhorou moderadamente, sendo que o melhor resultado obtido foi novamente através da métrica *Cosine Similarity* com recurso a *TF-IDFVectorizer*, mas neste caso utilizando trigramas.

No entanto, não sendo este desempenho muito satisfatório para a escolha de um candidato, foram testadas as mesmas métricas, mas para uma *Accuracy* diferente, sendo esta a *Accuracy* que estas métricas têm de acertar nos candidatos a excluir, ou seja, descartar os maus candidatos.

Para isso foi utilizada novamente a matriz de confusão média e, para este caso, foi necessário calcular a divisão do primeiro índice da primeira linha, que representa a coincidência da classificação por ambas as partes como sendo CVs maus, pela soma de toda essa linha, representando o total de CVs classificados como maus. Obtendo, assim, os resultados da tabela abaixo (Tabela 12):

Tabela 12 – Análise da Accuracy em exclusões das diferentes métricas utilizando n-grams

	Euclidean Distance c/ *TF-IDF*Vec	*Euclidean Distance* c/ CountVec	*Cosine Similarity* c/ *TF-IDF*Vec	*Cosine Similarity* c/ CountVec	*Jaccard Similarity*
Unigramas	57.40%	53.70%	57.40%	56.48%	50.43%
Bigramas	52.77%	50.92%	52.77%	54.62%	
Trigramas	60.18%	52.77%	62.96%	59.73%	
Uni + Bigramas	60.18%	53.70%	60.18%	56.48%	
Bi + Trigramas	52.77%	50.92%	53.70%	54.62%	
Uni + Bi + Trigramas	60.18%	53.70%	60.18%	55.56%	

Analisando esta tabela é possível verificar que todos os resultados são superiores aos anteriores, ou seja, as métricas demonstram uma melhor capacidade de categorizar um CV como sendo mau, e com a métrica *"Cosine Similarity"* com recurso a *TF-IDFVectorizer* novamente a obter os melhores resultados. Esta capacidade pode ser explicada devido ao facto de num CV classificado como mau existirem poucas ou quase nenhumas palavras em comum com o documento comparado, o que contribui também para a exclusão do CV por parte de um recrutador, i.e., se um candidato apresenta poucos ou quase nenhuns aspetos em comum com a Job ID mais facilmente será rejeitado pelo respetivo recrutador.

Posto isto, e visto que a *Accuracy* para exclusão de candidatos maus apresentou resultados mais satisfatórios que todos os anteriores, foi necessário observar qual a percentagem de bons candidatos excluídos, de entre todos os excluídos, obtendo a tabela que se segue (Tabela 13).

Tabela 13 – Análise de bons candidatos excluídos

	Euclidean Distance c/ *TF-IDF*Vec	*Euclidean Distance* c/ CountVec	*Cosine Similarity* c/ *TF-IDF*Vec	*Cosine Similarity* c/ CountVec	*Jaccard Similarity*
Unigramas	33.33%	43.13%	33.33%	39.21%	44%
Bigramas	43.13%	43.13%	33.33%	31.37%	
Trigramas	29.41%	39.21%	21.56%	22.43%	
Uni + Bigramas	33.33%	41.17%	33.33%	39.21%	
Bi + Trigramas	41.17%	43.13%	33.33%	31.37%	
Uni + Bi + Trigramas	33.33%	41.17%	33.33%	39.21%	

Mais uma vez a mesma métrica obteve os melhores resultados como pode ser observado na Tabela 13, no entanto esta percentagem, 21.56% de bons candidatos

excluídos na melhor das hipóteses, ainda se encontra muito alta relativamente ao assunto que trata.

6.2. Classificação automática através da utilização *keywords*

Visto a Job ID ser um ficheiro com grande quantidade de informação decidiu-se testar a mesma abordagem que no ponto 6.1), no entanto, em vez da utilização das Job ID's de cada departamento, foram retiradas *keywords* pertinentes de cada uma delas, calculando assim a similaridade entre os CVs e os ficheiros de *keywords*, sendo esta a única diferença entre esta abordagem e a anterior.

Foi novamente calculada a *Accuracy* para esta abordagem, da mesma forma que anteriormente.

Tabela 14 – Análise da Accuracy das diferentes métricas utilizando keywords

Métrica Utilizada	*Accuracy*	% de bons candidatos excluídos	*Accuracy* em exclusões
Euclidean Distance c/ *TF-IDF*Vec	50.57%	34.17%	58.33%
Euclidean Distance c/ CountVec	46.49%	43.13%	53.70%
Cosine Similarity c/ *TF-IDF*Vec	51.75%	33.33%	60.38%
Cosine Similarity c/ CountVec	50.09%	35.29%	55.55%
Jaccard Similarity	44.63%	38%	53.54%

Analisando os resultados presentes na Tabela 14 observa-se que existiram mudanças na *Accuracy* para esta nova abordagem, nomeadamente o aumento da *Accuracy* dos modelos utilizando *"Euclidean Distance"* e "*Cosine Similarity",* sendo que o último apresenta um aumento mais significativo face ao primeiro, cerca de 3%.

Os resultados acima demonstrados foram obtidos através da utilização de um conjunto de *n-grams* entre 1 e 3, e por essa razão foi também testado para esta nova implementação

a utilização dos mesmos conjuntos que na abordagem do ponto 6.1), unigramas, bigramas, trigramas, isoladamente e ainda, unigramas e bigramas, e por fim bigramas e trigramas.

No entanto, a implementação dos vários conjuntos de n-grams em nada influenciou a *Accuracy* para esta abordagem, o que era de esperar pois os ficheiros de keywords apenas contêm unigramas, logo não pode haver similaridade com unigramas, bigramas ou trigramas. Tal como na anterior abordagem foi também esta abordagem testada face à *Accuracy* destas métricas para a exclusão de candidatos e para a percentagem de bons candidatos excluídos (Tabela 14)

Novamente houve mudanças nesta *Accuracy*, como é possível observar na Tabela 14, em comparação à implementação do ponto 6.1), no entanto essas mudanças são negativas, o mesmo acontece para a percentagem de bons candidatos excluídos.

Concluindo esta abordagem, como podemos ver pela análise dos resultados, não existem melhorias significativas devido à implementação da similaridade de documentos apenas com *keywords*, isto deve-se ao facto de serem apenas comparadas palavras entre documentos e não expressões implementadas noutras abordagens através de *n-grams.*

6.3. Sumário

Neste capítulo foram apresentadas abordagens completamente automáticas no que diz respeito à classificação do CV de um candidato, com recurso a métricas de similaridade de texto em conjunto com ferramentas de aprendizagem automática para tratamento de texto. Numa primeira abordagem – 6.1) – foi tratado o caso da similaridade do texto presente num CV e uma determinada Job ID. De seguida – 6.2) – foi estudado o caso da similaridade do texto presente num CV e um conjunto de *keywords* descritivas de uma Job ID. Para ambas as abordagens foram calculadas duas *Accuracys*, sendo a primeira a *Accuracy* das diversas métricas de acertarem nos candidatos considerados como "Bons", "Maus" e "Medianos", e a segunda de acertarem apenas nos candidatos a excluir, ambas calculadas com recurso à classificação dada pelos recrutadores. Os melhores resultados foram obtidos logo na primeira abordagem – 6.1) – através do cálculo da segunda *Accuracy*, com recurso a *Cosine Similarity* em conjunto com *TF-IDFVectorizer*, obtendo 62.96%, contrastando com os resultados obtidos através do cálculo do primeiro, resultados estes que apresentam 54.76% como o melhor resultado, sendo que a melhor métrica para este caso foi a mesma que no caso anterior. Na segunda abordagem deste

capítulo os resultados não apresentaram qualquer melhoria, pelo contrário, observando os resultados obtidos para esta abordagem verifica-se que as precisões apresentam piores resultados. O principal fator que contribui para estes resultados deve-se ao facto de nesta abordagem não serem implementados conjuntos de *n-grams* superiores a um, conjuntos esses que apresentam os melhores resultados na primeira abordagem, nomeadamente trigramas.

Concluindo, neste capítulo é observado que ambos os classificadores automáticos criados apresentam melhores resultados aquando da classificação de um candidato como sendo "Mau", em vez de "Bom", no entanto, a primeira abordagem, devido ao facto de utilizar *n-grams* e uma Job ID composta por frases, supera em todos os aspectos a segunda, na qual apenas são utilizadas *keywords*, e por essa razão não ser possível implementar *n-grams.*

Capítulo 7. Conclusão

Um dos principais objetivos desta dissertação consiste em, recorrendo a estratégias de TM, identificar automaticamente quais os candidatos adequados a um determinado cargo, com base no conteúdo presente no seu CV. Importa realçar que a informação que consta de um CV é sobretudo informação não estruturada, baseada em texto, o que requer estratégias de processamento de língua natural, com vista a extrair daí o conhecimento necessário para uma tomada de decisão. Numa primeira fase foi realizada uma revisão da literatura com o intuito de estudar diversos casos realizados sobre a mesmas temáticas e as respetivas abordagens. Após esse estudo foram implementadas várias abordagens utilizando diversas técnicas de TM que se acharam pertinentes para esta dissertação.

Nas primeiras abordagens a tentativa foi de classificar todas as linhas presentes num CV de acordo com a sua categoria, de modo a segmentar um CV por blocos de categoria, para que fosse mais fácil retirar informação pertinente de cada um e, por conseguinte, implementar um método de classificação de um CV. Para todas estas abordagens foram consideradas quatro categorias: “Informação Pessoal”,” Educação”, “Experiência Profissional” e “Outros”, as três primeiras devido ao facto de serem as mais pertinentes aquando da avaliação de um CV, e a quarta para englobar informação que não pertence a qualquer uma das três anteriores. Essas abordagens não apresentaram resultados satisfatórios pelo que a classificação automática de um CV não foi possível por via das mesmas.

Devido a isso, de seguida foram implementadas diversas abordagens para classificação de um CV completamente automáticas. Foram utilizadas diversas métricas para estas abordagens todas com resultados similares.

A classificação automática de um CV como sendo bom, mau ou mediano foi possível, no entanto não foram apresentados resultados muito precisos. Com recurso às diversas métricas foi possível observar que estas apresentam melhores resultados quando utilizadas para a exclusão de maus candidatos, o que pode contribuir para uma nova abordagem ao problema da classificação automática de CVs, ao excluir os candidatos indesejados apenas seria necessário analisar CVs de candidatos bons e medianos, contribuindo assim para uma poupança de tempo e recursos significativa.

Finalmente para este estudo conclui-se que é possível a implementação de um classificador automático de CVs, no entanto nunca com o grau de precisão de um humano, neste caso um gestor de recursos humanos.

Bibliografia

Baxter, R., Hastings, N., Law, A., & Glass, E. J. . (2008). Text Mining using Non-Negative Matrix Factorizations. *Animal Genetics*, *39*(5), 561–563.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2000). Latent Dirichlet Allocation. *CrossRef Listing of Deleted DOIs*, *1*(1), 993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993

Cavnar, W. B. and T. J. M. (1994). N-gram based text categorization. *Proceedings of Symposium on Document Analysis and Information Retrieval*, 161–175.

Drake, J. (2002). Python Dictionaries. Retrieved from http://www.afterhoursprogramming.com/tutorial/Python/Dictionaries/

Faliagka, E., Iliadis, L., Karydis, I., Rigou, M., Sioutas, S., Tsakalidis, A., & Tzimas, G. (2014). On-line consistent ranking on e-recruitment: Seeking the truth behind a well-formed CV. *Artificial Intelligence Review*, *42*(3), 515–528. https://doi.org/10.1007/s10462-013-9414-y

Faliagka, E., Tsakalidis, A., & Tzimas, G. (2012). An integrated e-recruitment system for automated personality mining and applicant ranking. *Internet Research*, *22*(5), 551–568. https://doi.org/10.1108/10662241211271545

G. D. Forney, J. (1973). The Viterbi Algorithm. *Proc. IEEE*, *61*, 268–278.

H.Gomaa, W., & A. Fahmy, A. (2013). A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, *68*(13), 13–18. https://doi.org/10.5120/11638-7118

Huang, A. (2008). Similarity measures for text document clustering. *Proceedings of the Sixth New Zealand*, (April), 49–56. https://doi.org/10.1109/ICDMW.2009.61

Joachims, T. (1996). A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization.pdf.

Kessler, R., Torres-Moreno, J. M., & El-Bèze, M. (2007). E-Gen: Automatic Job Offer Processing System for Human Resources. In *MICAI 2007: Advances in Artificial Intelligence* (pp. 985–995). https://doi.org/10.1007/978-3-540-76631-5

Landauer, T. K., Folt, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*(2), 259–284. https://doi.org/10.1080/01638539809545028

Lee, D., & Seung, H. (2001). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, (1), 556–562. https://doi.org/10.1109/IJCNN.2008.4634046

Medina, C. P., & Ramon, M. R. R. (2015). How to learn professional competencies via blogs. *New Educational Review*, *42*(4), 40–51. https://doi.org/10.15804/tner.2015.42.4.03

Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and Knowledge-based Measures of Text Semantic Similarity. *AAAI'06 Proceedings of the 21st National Conference on Artificial Intelligence*, 775–780. https://doi.org/10.1.1.65.3690

Pedregosa, F., & Varoquaux, G. (2011). *Scikit-learn: Machine learning in Python*. *… of Machine Learning …*. https://doi.org/10.1007/s13398-014-0173-7.2

Quan, X., Liu, G., Lu, Z., Ni, X., & Wenyin, L. (2010). Short text similarity based on probabilistic topics. *Knowledge and Information Systems*, *25*(3), 473–491. https://doi.org/10.1007/s10115-009-0250-y

Suphakit Niwattanakul*, Jatsada Singthongchai, E. N. and S. W. (2013). Using of Jaccard Coefficient for Keywords Similarity. *Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong*, *Vol I*(May 2017), 380–384. https://doi.org/ISBN 978-988-19251-8-3

Tjong, E. F., Sang, K., & Meulder, F. De. (2003). Utsunomiya_1996_PA365.pdf, 142–147.

Tosik, M., Rotaru, M., Goossen, G., & Hansen, C. L. (2015). Word Embeddings vs Word Types for Sequence Labeling : the Curious Case of CV Parsing. *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 123–128. https://doi.org/10.3115/v1/W15-1517

W. Ling, C. Dyer, AW. Black, I. T. (2005). An Evaluation of Home-Based Telemedicine Services. *Https://Clinicaltrials.Gov/Show/Nct00105846*, 1299–1304. https://doi.org/10.1016/j.jpedsurg.2006.03.053

Yih, W.-T., Toutanova, K., Platt, J., & Meek, C. (2011). Learning discriminative projections for text similarity measures. *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, (June), 247–256. Retrieved from http://dl.acm.org/citation.cfm?id=2018965

Yu, K., Guan, G., & Zhou, M. (2005). Resume Information Extraction with Cascaded

Hybrid Model. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, (June), 499–506. https://doi.org/10.3115/1219840.1219902

ZhiXiang, J., Chuang, Z., Bo, X., & ZhiQing, L. (2009). Research and implementation of intelligent chinese resume parsing. In *Proceedings - 2009 WRI International Conference on Communications and Mobile Computing, CMC 2009* (Vol. 3, pp. 588–593). https://doi.org/10.1109/CMC.2009.253