Resposta Contextual Automática a Mensagens Electrónicas

Pedro José de Oliveira

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
Área de Especialização em
Tecnologias da Decisão e Conhecimento

Orientador: Doutor João Manuel Simões da Rocha

Júri:
Presidente:
Doutora Maria de Fátima Coutinho Rodrigues, Professor Coordenador
Vogais:
Doutor João Manuel Simões da Rocha, Professor Coordenador
Doutora Maria João Monteiro Ferreira Viamonte, Equiparado Professor Adjunto

Porto, Outubro 2009

Agradecimentos

Esta dissertação de Mestrado não teria sido possível sem a colaboração de algumas pessoas às quais gostaria de exprimir os meus agradecimentos:

Ao Dr. João Rocha, orientador da dissertação, agradeço o apoio, a partilha de saberes e a disponibilidade demonstrada em todas as fases que levaram à concretização deste trabalho;

À Eng. Berta Baptista pela disponibilidade no fornecimento das mensagens de correio electrónico, indispensáveis para o desenvolvimento da dissertação;

À Comissão Directiva do Departamento de Engenharia Informática do ISEP pelo apoio dado na impressão da dissertação;

A todos os meus colegas do Departamento de Engenharia Informática do ISEP que ao longo deste último ano me deram apoio para a conclusão deste projecto;

À minha família pela motivação, compreensão e força, com que sempre me acompanharam;

Finalmente, à minha companheira Mariana Oliveira.

Resumo

A evolução tecnológica, associada às mudanças sociais a que temos assistido, nomeadamente nas últimas décadas, originou mudanças significativas na forma como os utentes interagem com as instituições, passando a privilegiar a utilização de meios electrónicos, tais como as mensagens de correio electrónico, em detrimento de formas mais tradicionais, como a carta e o telefone. Neste contexto, sendo o ISEP uma instituição de ensino superior que alberga milhares de alunos e recebe centenas de novos alunos todos os anos, necessita de ter condições para que possa responder de forma atempada às inúmeras mensagens de correio electrónico que recebe. Esta necessidade fez com que surgisse um projecto, de nome SiRAC, que servisse para auxiliar na resposta a essas mensagens.

O SiRAC tem como objectivo responder a mensagens de correio electrónico de forma automática. De salientar que se admite não ser possível responder a todas as mensagens, privilegiando-se aquelas que são recorrentemente colocadas à Divisão Académica. Assim será possível encurtar o tempo de comunicação entre os diversos intervenientes, criando uma relação mais próxima entre o ISEP e o público que o contacta. O SiRAC analisa as mensagens e procura responder de forma automática sempre que o seu conteúdo possa ser classificado como fazendo parte de um conjunto de questões previamente identificadas pelos recursos humanos da Divisão Académica como recorrentes e para as quais já exista uma resposta tipo. As questões constantes da mensagem são identificadas através de palavras e expressões normalmente associadas aos diferentes tipos de questão. O envio da resposta

pressupõe a identificação correcta dos tipos associados e de acordo com requisitos mínimos definidos, de forma a evitar enviar uma resposta errada a uma mensagem.

A implementação do SiRAC permite a libertação de recursos humanos da Divisão Académica que anteriormente estavam afectas à resposta de mensagens para o desempenho de outras funções.

Palavras-chave: Sistema de resposta automática; Processamento de linguagem natural; Algoritmos de classificação.

Abstract

Technological developments, linked to social changes that we have assisted, particularly in recent decades led to significant changes in the way users interact with institutions, focusing on the use of electronic means such as e-mail messages, in detriment of more traditional forms such as letter and phone. In this context, ISEP as a higher education institution which is attended by thousands of students and receives hundreds of new students each year needs to be able to respond in a timely manner to the numerous e-mails it receives. This need made a project arise, named SiRAC, which will assist in responding to these messages.

SiRAC is designed to respond to e-mail messages automatically. Note that we admit that it is not possible to answer to all messages, focusing on those that are repeatedly placed to the Academic Services. This way it is possible to shorten the communication time between the various intervenient, creating a closer relationship between ISEP and the public that contacts them. SiRAC analyzes the e-mail and attempts to answer automatically when its contents can be classified as part of a set of queries already identified by the human resources of the Academic Services and for which there is already a response type. The questions contained in the e-mail are identified by words and expressions usually associated with different types of questions. The response requires the correct identification of associated types and in accordance with minimum requirements set out in order to avoid sending a wrong answer to a message.

The implementation of SiRAC allows the release of human resources at the Academic Services who were previously engaged in responding messages to perform other tasks.

Keywords: Automatic response systems; Natural language processing; Classification algorithms.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Gráficos

1. Introdução

A forma de comunicar tem vindo a alterar-se muito nas últimas décadas. Desde o século XIX que a comunicação escrita tem evoluído, alterando os hábitos do ser humano. Nos finais do século XIX utilizava-se o telégrafo para enviar mensagens curtas, passando-se para a massificação da utilização da carta tradicional no século XX evoluindo-se na actualidade para a utilização intensiva de correio electrónico, vulgarmente conhecido por email.

Esta rápida evolução, desde a Revolução Industrial, é um reflexo da sociedade moderna, dinâmica e exigente, que procura soluções eficientes e eficazes para os seus problemas num curto espaço de tempo.

Desta forma não surpreende que, a evolução da utilização do correio electrónico nos últimos vinte anos, tenha levado a humanidade a tentar implementar melhorias no envio, transporte e recepção de mensagens de correio electrónico, pois constitui uma das formas de comunicação mais utilizadas por ser simples, rápida e de baixo custo.

Não é de estranhar que grandes empresas e instituições mundiais privilegiem este tipo de comunicação para a interacção com os seus clientes/utentes. Para que seja eficaz, torna-se necessário que as respostas sejam atempadas, o que nem sempre se torna fácil devido à grande quantidade de mensagens de correio electrónico recebidas. A solução mais simples é a de aumentar os recursos

humanos dedicados a esta tarefa, com os custos que lhe são inerentes e sem que se garanta um aumento de produtividade global.

Um bom exemplo são as instituições de ensino superior que diariamente recebem dezenas, senão centenas de mensagens, as quais necessitam de resposta atempada para que as expectativas de utilização do correio electrónico não sejam defraudadas quando comparadas com a utilização dos meios tradicionais.

Neste contexto o Instituto Superior de Engenharia do Porto (ISEP), procura implementar um sistema que permita uma resposta automática a um número significativo das mensagens de correio electrónico recepcionadas na Divisão Académica, de forma a conseguir minimizar o tempo de resposta e a libertar os recursos humanos para outras actividades. Refira-se que o número de mensagens recebidas não é constante ao longo do ano, sendo maior quando a necessidade de recursos humanos para outras tarefas é também maior, podendo atingir mais de um milhar nos períodos de pico.

Após análise dum número significativo de mensagens enviadas para a Divisão Académica foi possível verificar a existência de perguntas recorrentes, possibilitando a tipificação de respostas, segundo temas/tipos. Mas, apesar de a interacção ser feita entre uma instituição de ensino superior e alunos/candidatos/ex-alunos, nem sempre a construção frásica está de acordo com as regras gramaticais, havendo também falhas do ponto de vista semântico. Tudo isto traduz-se em mensagens com perguntas mal formuladas, vagas e incorrectas, o que dificulta a análise do seu conteúdo (por vezes, mesmo para um ser humano, é praticamente impossível perceber o que o remetente realmente pretende). Estes serão à partida os grandes desafios para o desenvolvimento de um sistema com estas características.

A dissertação que agora se apresenta tem como objectivo o desenvolvimento de um sistema que tentará responder de forma automática e baseada no contexto, a um conjunto significativo de mensagens de correio electrónico, de acordo com tipos definidos *a priori*, recebidas na caixa de correio electrónico da Divisão Académica do ISEP.

A dissertação está organizada da seguinte forma:

- Estado da arte dos diferentes conceitos utilizados no desenvolvimento do projecto;
- Apresentação e caracterização do projecto SiRAC;
- Conclusões retiradas dos testes efectuados;
- Trabalho futuro.

2. Estado da arte

2.1. Introdução

O sistema proposto será constituído por módulos distintos que necessitarão de ser desenvolvidos e que serão utilizados de forma sequencial no processamento das mensagens.

Assim, no servidor responsável pela recepção das mensagens de correio electrónico, será necessário interceptar as mensagens com destino à Divisão Académica do ISEP para a extracção do seu conteúdo. Inicialmente esse conteúdo será alvo de uma análise para verificar a existência de erros ortográficos, com a sua consequente correcção, através de abordagens apresentadas a seguir, de forma a possibilitar o reconhecimento da totalidade das palavras constantes da mensagem de correio electrónico.

Na fase seguinte, o sistema através de técnicas de pergunta e resposta (*question answering*) irá interpretar as questões colocadas em linguagem natural pelo remetente, utilizando algoritmos de classificação, de forma a concluir o que realmente é pretendido, validando a questão colocada num dos temas/tipos previamente definidos.

Após a definição do tipo/tema de questão, seguir-se-á a construção da resposta que poderá ser uma resposta simples (respostas pré-definidas) ou mais complexa (construção dinâmica, baseado nas respostas pré-definidas), conforme a análise efectuada.

No final, o sistema deverá enviar uma mensagem de correio electrónico ao remetente com uma resposta concisa e com informação suficiente para satisfazer as suas dúvidas.

Desta forma, neste capítulo será dado destaque a três temáticas que estão subjacentes ao desenvolvimento desta dissertação:

- Correcção automática de erros ortográficos – neste subcapítulo iremos apresentar uma temática, com algumas dezenas de anos de investigação (comprovada com algumas referências apresentadas ao longo do texto), dando destaque às técnicas mais utilizadas/divulgadas;
- Construção automática de respostas – neste subcapítulo será apresentado o conceito de sistemas de pergunta e resposta (*question answering*) uma especialização dos sistemas de recuperação de informação (*information retrieval*);
- Algoritmos de classificação – este subcapítulo focaliza-se no algoritmo *Naïve Bayes*, devido à sua grande utilização em situações de classificação e com resultados bastante bons.

2.2. Correcção automática de erros ortográficos

2.2.1. Introdução

A necessidade do desenvolvimento de algoritmos e técnicas para corrigir automaticamente palavras no texto tem revelado constituir um grande desafio. A investigação nesta área iniciou-se há já algumas décadas e tem tendência para continuar devido à complexidade do tema. Os algoritmos e as técnicas propostas para a correcção automática de palavras são definidos de acordo com os tipos de correcção que se pretendam realizar.

A verificação e correcção automática de texto, através de dispositivos computacionais, ultrapassa o simples objectivo de auxiliar na edição de texto. Algoritmos e técnicas com este objectivo têm

utilidade, por exemplo, em aplicações de reconhecimento da fala e caracteres (OCR[1]) (Almeida, et al., 2004).

A identificação de um erro num texto é diferente da sua correcção. Identificar implica perceber e apontar o erro existente. A correcção de uma palavra num texto, de forma automática, é um problema bastante mais complexo do que apenas identificar um erro, sendo necessário encontrar outra palavra que substitua a errada.

O problema de corrigir uma palavra escrita incorrectamente para a palavra originalmente pensada por quem escreve tem interessado muitos investigadores. No entanto, apesar da investigação, os melhores correctores automáticos ainda não conseguiram ultrapassar os 96% de acerto no retorno da palavra correcta como a melhor candidata de correcção, apesar de níveis mais elevados serem alcançados para encontrar a palavra correcta, entre uma lista de possíveis correcções. Desta forma, qualquer corrector ortográfico razoável deve ser interactivo, ou seja, deve apresentar ao utilizador uma lista de possíveis correcções, e deixar a decisão final sobre qual a palavra a escolher para o utilizador. Muitas horas de trabalho humano poderiam ser poupadas se a taxa de precisão para a palavra melhor candidata a correcta fosse melhorada. Com uma taxa de precisão melhorada, os correctores automáticos poderiam passar de interactivo para automático, ou seja, poderiam funcionar sem intervenção humana (Bodine, 2005).

A investigação destinada a corrigir as palavras em texto concentrou-se em três problemas progressivamente mais difíceis: (Kukich, 1992)

1. Identificação de erros em palavras (*nonword error detection*);
2. Correcção de palavras isoladas (*isolated-word error correction*);
3. Correcção dependente de contexto (*context-dependent word correction*).

Em resposta ao primeiro problema, um padrão de correspondência eficiente e técnicas de análise n-grama têm sido desenvolvidos para detectar *strings*[2] que não aparecem num determinado dicionário. Em resposta ao segundo problema, uma variedade de técnicas de correcção de erros ortográficos de aplicação geral e específica foram desenvolvidos. Alguns deles foram baseados em estudos detalhados de padrões de erros ortográficos. Em resposta ao terceiro problema, algumas experiências utilizando ferramentas de processamento de língua natural ou modelos estatísticos de linguagem foram realizadas.

[1] **OCR** é um acrónimo para o inglês *Optical Character Recognition*, uma tecnologia para reconhecer caracteres a partir de um arquivo de imagem ou mapa de bits.
[2] Conjuntos de caracteres

A seguir serão apresentados os conceitos relacionados com os problemas, descrevendo algumas técnicas propostas para solucionar o problema de detecção e correcção de erros em textos.

2.2.2. Identificação de erros em palavras

O objectivo desta técnica consiste em identificar quando uma determinada palavra não faz parte de um dicionário pré-existente.

As décadas de 70 e 80 corresponderam a um período em que se desenvolveram trabalhos nesta área, envolvendo principalmente técnicas para reconhecimento de padrões e comparação de cadeias de caracteres para verificação da ausência de uma palavra num conjunto de palavras ou dicionário pré-existente (Kukich, 1992).

O factor mais importante e crítico para detectar um erro numa palavra isolada é a utilização de técnicas eficientes de pesquisa num dicionário. Além disso, a manutenção das palavras que compõem o dicionário é uma questão de extrema importância, devendo este ser composto por um número razoável de palavras comummente encontradas em textos para que a verificação não identifique frequentemente detecções falsas. No caso contrário, um número demasiado grande de palavras pode levar a que alguns erros não sejam detectados.

Por outro lado, esta técnica tem as suas limitações na identificação de erros nas palavras quando utilizada isoladamente. Tomando a palavra *consciência* como exemplo, supondo que existe o verbo *conscienciar* e que o dicionário do corrector ortográfico contém as formas deste verbo, sempre que o utilizador se esquecer de colocar o acento em *consciência*, o corrector não será capaz de detectar este erro, pois reconhecerá a palavra como sendo uma forma do verbo *conscienciar*. Seria preferível que o corrector não reconhecesse a palavra *consciencia*, e assinalasse erro quando a encontrasse, do que deixar passar em branco todos os outros erros por falta de acento (Medeiros, 1995).

Este problema leva a um outro, intrínseco aos correctores ortográficos: fazem a correcção ortográfica palavra a palavra, fora de contexto. Assim, nunca poderão detectar utilizações incorrectas de palavras correctamente escritas. Por exemplo, no caso da utilização da palavra *numero* em vez de *número*, ou *poças* em vez de *possas*, o corrector nunca será capaz de detectar o erro sem que seja feita uma análise contextual.

De seguida serão apresentadas duas técnicas relevantes e de utilização frequente:

- **Pesquisa em dicionário**

A pesquisa em dicionário é uma tarefa simples. No entanto, o tempo de resposta torna-se um problema quando a dimensão do dicionário excede as centenas de palavras. No processamento de documentos e na recuperação de informação (*information retrieval*), o número de entradas no dicionário pode variar entre 25.000 e 250.000 palavras. Este problema tem sido abordado de três formas (Kukich, 1992):

1. Pesquisa eficiente em dicionário e/ou algoritmos de reconhecimento de padrões;
2. *Schemas* de particionamento do dicionário;
3. Técnicas de processamento morfológico.

A técnica mais comum para implementar a pesquisa em dicionário é a utilização de uma tabela *hash*. Segundo esta técnica, para pesquisar uma palavra fornecida, o sistema calcula o seu endereço de *hash* (de uma tabela pré-construída), através de uma transformação aritmética que retorna a palavra guardada nesse endereço. Se a palavra guardada no endereço da *hash* for diferente da palavra fornecida ou for *null*, é indicada a existência de um erro ortográfico. Ou seja, os registos guardados numa tabela são directamente endereçados a partir de uma transformação aritmética sobre a chave de pesquisa (Ziviani, 2004).

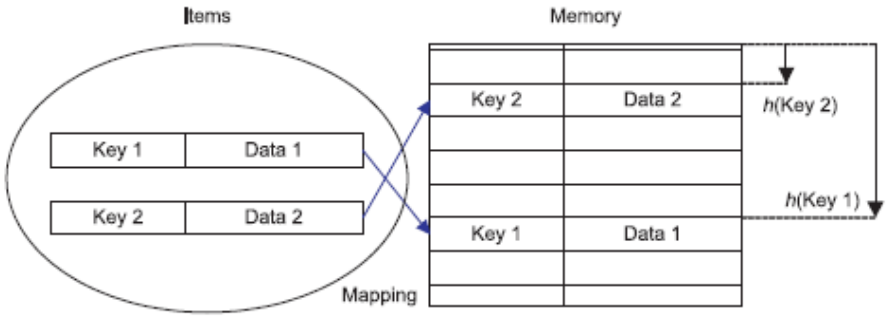

Figura 1 – Mapeamento de chave de pesquisa (palavra) nos registos da tabela, utilizando uma função (*h(.)*) *hash* (Cho, et al., 2007)

Turba (Turba, 1981) apresenta uma pequena revisão dos prós e contras do uso de tabelas *hash* para a pesquisa em dicionário. A principal vantagem é que a natureza de acesso aleatório de um código *hash* elimina um grande número de comparações necessárias para pesquisas sequenciais ou em árvore sobre o dicionário. A principal desvantagem é a necessidade de conceber uma função *hash* inteligente que evite colisões e sem exigir uma enorme tabela *hash*.

- **Análise de n-grama**

  Um n-grama é uma sequência de ***n*** letras de uma palavra, em que ***n*** é geralmente igual a 1, 2 ou 3, sendo respectivamente identificado por um monograma, bigrama ou trigrama. Segundo esta técnica é examinado cada n-grama de uma palavra de entrada e de seguida consultada uma tabela de n-gramas válidos. Na língua portuguesa, temos como exemplos inválidos os n-gramas: np, nb e qrl. Para a execução desta técnica é necessário o preenchimento prévio da tabela de n-gramas (uma forma bastante utilizada é a partir do pré-processamento de um dicionário) (Almeida, et al., 2004)(Kukich, 1992).

  Segundo (Almeida, et al., 2004), através de uma matriz de 26X26 é possível chegar à forma simples de um bigrama. Desta forma, uma posição da matriz representa a combinação de duas letras do alfabeto; possuindo o valor 1 (um) quando a combinação entre as letras é válida para pelo menos uma das palavras que compõem o dicionário, possuindo o valor 0 (zero) no caso contrário.

  Para exemplificar, a palavra “laranja” pode ser dividida em cinco trigramas: “lar”, “ara”, “ran”, “anj” e “nja”. Normalmente, os erros ortográficos mais comuns afectam poucos constituintes de n-grama, o que nos leva a procurar pela palavra correcta através daqueles que compartilham a maior parte dos n-gramas com a palavra errada.

2.2.3. Correcção de palavras isoladas

As técnicas desenvolvidas com este objectivo identificam e proporcionam a correcção de erros em palavras isoladas no texto. Os primeiros estudos relativos a esta área remontam à década de 60 do século passado.

Nalgumas ferramentas, a correcção é realizada com a intervenção do utilizador. Noutras circunstâncias, porém, pode ser necessário que a correcção seja automática; neste caso, a própria ferramenta decide como será corrigido o erro.

As técnicas para esse tipo de correcção seguem geralmente três passos: detecção do erro, geração de correcções candidatas e ordenação das correcções candidatas. Algumas vezes, este último passo é dispensado. Se por um lado interessa que a lista de correcções candidatas seja suficientemente grande para incluir a sugestão correcta, por outro lado deve ser suficientemente pequena para que o utilizador não tenha que procurar a palavra correcta numa lista de palavras demasiado extensa. O ideal seria que a lista de correcções candidatas contivesse uma só palavra: a correcta. Não sendo tal possível, deve-se

tentar que a palavra correcta esteja entre as primeiras. Resumindo, a lista de correcções candidatas deve ser tão pequena quanto possível, desde que a palavra correcta esteja presente.

Através de estudos desenvolvidos ao longo dos anos foi possível concluir pela existência de padrões nos tipos de erros cometidos na edição de palavras: (Kukich, 1992)

- muitos erros são instâncias simples de inserção, exclusão, substituição ou transposição de caracteres, ou seja, existe uma tendência de apenas cometer um desses tipos de erro na edição de uma palavra;
- uma palavra escrita de forma errada normalmente tem apenas um caracter diferente da sua grafia correcta;
- a primeira letra da palavra apresenta normalmente poucos erros.

A seguir serão descritas duas técnicas para a resolução deste tipo de problema: (Almeida, et al., 2004)

- **Distância mínima de edição**

  Um importante algoritmo criado em 1964 introduziu o conceito de distância mínima de edição ou distância Levenshtein: "é a quantidade mínima de operações de edição (inserção, exclusão e substituição) necessária para transformar uma cadeia de caracteres noutra cadeia de caracteres". Em geral, algoritmos baseados neste conceito requerem ***m*** comparações entre a palavra e o dicionário, onde ***m*** é o número de entradas do dicionário. Através desta técnica, os melhores candidatos para a palavra correcta são aqueles que apresentam a mínima distância de edição (Almeida, et al., 2004)(Kukich, 1992).

  Por exemplo, a distância de edição entre "itens" e "itesm" é 2, já que com apenas duas edições conseguimos transformar uma palavra na outra, e não há maneira de o fazer com menos de duas edições:

  1. itesm (palavra inicial)
  2. itesn (substituição de m por n)
  3. itens (transposição de n por s – palavra final)

  Neste caso é necessário fazer uma substituição e uma transposição, não necessariamente por esta ordem. Temos então Ed(itens, itesm) = 2 (Pacheco, 1996).

A seguir é apresentado outro exemplo demonstrando que não existe apenas um *caminho* na aplicação desta técnica (a palavra *levenshtein* tem pelo menos dois *caminhos* que pode seguir na construção da palavra *meilenstein*):

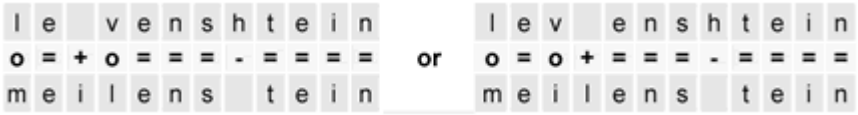

Figura 2 – Exemplo de aplicação da distância mínima de edição[3].

- **Similaridade de chaves**

O conceito de similaridade de chaves tem como objectivo mapear toda a palavra a analisar numa chave, para que palavras com grafias semelhantes tenham a mesma chave. A grande diferença e consequente vantagem desta técnica é que, para encontrar as correcções candidatas, não é necessário comparar a palavra com todas as palavras do dicionário. Apenas é necessário gerar a chave para a palavra errada e procurar as palavras com a mesma chave para uma lista de candidatos.

Kukich (Kukich, 1992) refere um exemplo antigo desta técnica: o sistema SOUNDEX (Odell, 1918) que mapeia uma cadeia de caracteres numa chave composta pela primeira letra da palavra, mais uma sequência de dígitos, determinada segundo a regra abaixo apresentada. Os zeros e as repetições de caracteres são descartados da chave resultante.

A, E, I, O, U, H, W, Y : 0
B, F, P, V : 1
C, G, J, K, Q, S, X, Z : 2
D, T : 3
L : 4
M, N : 5
R : 6

Após a aplicação destas regras, temos os seguintes exemplos: (Kukich, 1992)

- WEKK – W022 – W2;
- WEEK – W002 – W2;
- WEAK – W002 – W2.

Todos os exemplos terão a mesma chave, pois têm grafias semelhantes.

[3] Imagem retirada de: http://www.levenshtein.net/images/levenshtein_meilenstein_path.gif

2.2.4. Correcção dependente de contexto

As técnicas descritas anteriormente não conseguem resolver o problema de uma palavra escrita, sem qualquer erro, ser utilizada no lugar de outra, ou seja, não consideram a estrutura sintáctica em que a palavra foi utilizada.

Segundo Pacheco (Pacheco, 1996), a estrutura sintáctica da frase é das componentes mais importantes nos processos associados com a utilização de linguagem natural. Um problema relevante envolvido neste tipo de análise é a necessidade de colocar as palavras por categorias.

Para o tratamento deste tipo de erros, podemos utilizar duas abordagens distintas: (Kukich, 1992)

- **Uso de técnicas de Processamento de Linguagem Natural (PLN)**

  O Processamento de Linguagem Natural, segundo Peter Jackson (Jackson, 2002) é o conjunto de métodos formais para analisar textos e gerar frases escritas num idioma humano. Normalmente, os computadores estão capacitados para compreender instruções escritas em linguagens de programação como o *Java*, *C*, *Perl* e outras, mas possuem muita dificuldade em entender comandos escritos em linguagem humana/natural. Isto resulta do facto das linguagens de programação serem extremamente precisas, contendo regras fixas e estruturas lógicas bem definidas que permitem ao computador saber exactamente como deve proceder a cada comando. Por outro lado, em linguagem natural, uma simples frase pode conter ambiguidades, nuances e interpretações que dependem do contexto, do conhecimento do mundo, de regras gramaticais, culturais e de conceitos abstractos.

  Um dos objectivos do Processamento de Linguagem Natural é fornecer aos computadores a capacidade de entender e compor textos. E "entender" um texto significa reconhecer o contexto, fazendo as respectivas análises: léxico-morfológica, sintáctica, semântica, análise do discurso e processamento pragmático (conforme a figura seguinte).

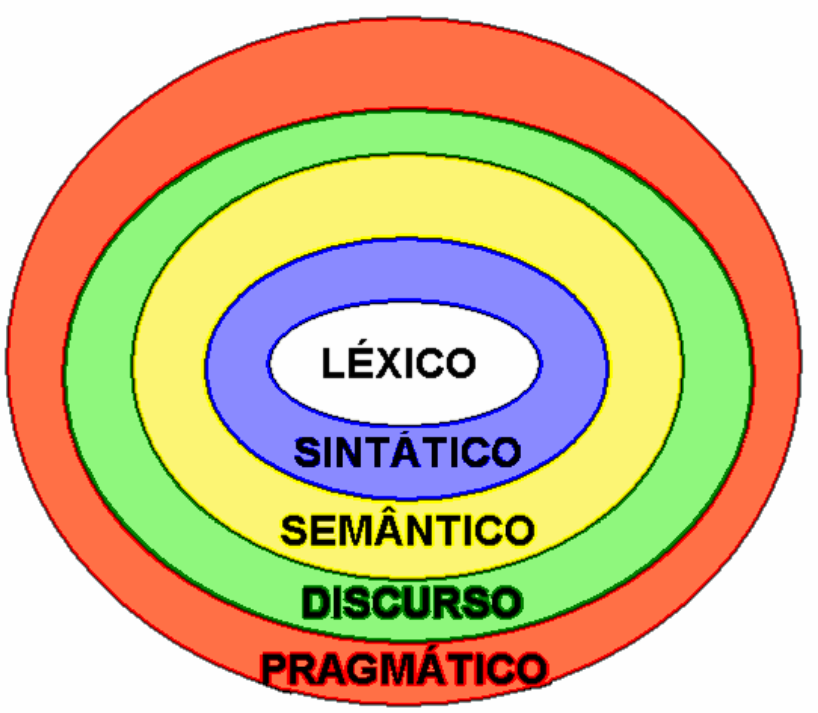

Figura 3 – Níveis de Processamento de Linguagem Natural (Von Wangenheim, 1993)

De acordo com as técnicas de PLN, considera-se que existe um erro numa palavra quando uma das regras do PLN foi violada, sendo então aplicadas ferramentas de PLN para procurar e corrigir o erro.

Serão de seguida descritas as fases necessárias para um sistema computacional interpretar uma frase em Linguagem Natural (Vinhaes, 2005):

Análise Morfológica

A análise morfológica estuda a forma como as palavras e os grupos de palavras constituem os elementos que expressam um idioma. A morfologia trata todo o conhecimento sobre a estrutura da palavra. Podemos verificar a existência de palavras que não podem ser particionadas, como “árvore”, mas tal é possível com as palavras “árvores”, “arvorezinhas” ou “impossível”. Os constituintes podem ser independentes como na palavra “árvore” ou dependentes como no caso do sufixo “zinhas” em “arvorezinhas” ou do prefixo “im” em “impossível” (Vieira, et al., entre 2002 e 2004).

O analisador morfológico tem como função a identificação num texto de palavras ou expressões isoladas. Estas palavras são classificadas de acordo com a sua categoria gramatical especificada em linguagem natural.

A finalidade da análise morfológica consiste em separar o texto em *tokens*[4], que são elementos essenciais para identificação dos termos prefixos, sufixos e formação das raízes que compõem a língua em questão. Para que seja possível o reconhecimento dos *tokens*, é necessário que estes sejam reconhecidos pelo conjunto de palavras (dicionário) que constituem o idioma em questão (Specia, 2000).

Se um *token* analisado não estiver dentro da estrutura léxica analisada, será retornado um erro, interrompendo o processo de compreensão da frase. O exemplo seguinte demonstra como é reconhecida uma frase em linguagem natural:

Exemplo 1: "Eu quero imprimir o arquivo.init do Mário"

Segundo Specia (Specia, 2000), na análise morfológica são realizados os seguintes procedimentos:

- Primeiro separa-se a expressão "do Mário" no substantivo próprio "Mário" na preposição "de" e no artigo "o".
- O passo seguinte é reconhecer a sequência ".init" como um tipo de extensão de arquivo que está sob a forma de adjectivo na sentença.
- Finalmente são atribuídas as categorias sintácticas a todas as palavras da frase, pois as interpretações dos afixos podem estar dependentes de sua categoria sintáctica.

Análise Sintáctica

A análise sintáctica é um dos componentes de uma linguagem. É o modo como a língua está organizada. A gramática de qualquer idioma não deve estar atrelada a regras rígidas, uma vez que nenhum conjunto simples de regras pode discorrer todas as maneiras em que se dá a comunicação entre as palavras. A sintaxe trata da relação lógica das palavras numa frase.

Analisar sintacticamente uma frase significa decompô-la nos seus elementos constituintes (sujeito, predicado, entre outros), verificando assim a relação lógica existente entre esses elementos.

O analisador sintáctico utiliza a gramática da linguagem natural a ser analisada e cruza com as informações do analisador morfológico, construindo uma árvore de derivação para a frase analisada, em que são mostradas as relações entre as palavras.

[4] Sequência de caracteres delimitadas por caracteres primitivos como espaço (" "), vírgula, ponto etc.

Esta árvore de derivação tem como função converter a lista de palavras que formam a frase numa estrutura hierárquica representativa de cada palavra da frase separadamente (Oliveira, 2004).

Cada uma das unidades geradas, corresponde aos componentes que serão atribuídos aos significados quando realizada a análise semântica (apresentada a seguir). A função da análise sintáctica é diminuir a quantidade de componentes que a semântica terá que analisar, reduzindo a complexidade do sistema, considerando que o processamento sintáctico utiliza menos recursos computacionais do que o processamento semântico (Specia, 2000).

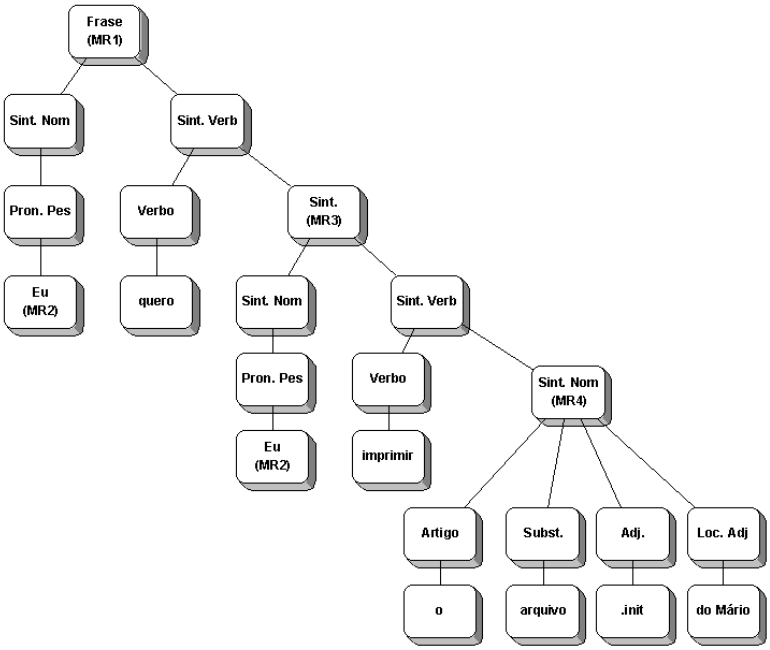

Figura 4 – Representação de uma árvore sintáctica (Oliveira, 2004)

Análise Semântica

A expressão análise semântica tem origem grega e significa dissecação (estudo minucioso). O objectivo desta análise é descobrir uma estratégia para representar o sentido das frases.

Após a definição do idioma, o processo de análise semântica irá procurar num dicionário (léxico), para extrair o significado das palavras que constituem as estruturas decompostas pelo analisador sintáctico (*parser*). Após este passo, as frases serão processadas como um todo, para verificação das regras semânticas, de acordo com o idioma em questão (Specia, 2000).

O analisador semântico tem como função verificar o sentido da estrutura das palavras que foram reagrupadas pelo analisador sintáctico, junto à árvore de derivação (árvore sintáctica), construída com as informações do analisador morfológico e sintáctico.

Segundo Oliveira (Oliveira, 2004), a análise semântica é feita para dar sentido às estruturas das palavras que foram reagrupadas pelo analisador sintáctico, uma vez que a análise morfológica já identificou cada uma delas individualmente.

Exemplo 2: "O passeio atropela timidamente."

Analisando o Exemplo 2 podemos verificar que estamos perante uma frase formada correctamente, sendo composta por um artigo ("O"), substantivo ("passeio"), verbo ("atropela") e advérbio ("timidamente"). No entanto não tem sentido! "Passeio" é um substantivo, mas não pode vir acompanhada do verbo atropelar, pois ele não atropela. "Atropelar" é um verbo, ao qual não se pode aplicar o advérbio "timidamente". Resumindo, é uma frase correcta do ponto de vista da sintaxe, mas, analisando semanticamente, constata-se que não tem sentido.

Para a análise semântica existir, é necessário que anteriormente tenha sido feita a análise sintáctica. Podemos concluir então, que o analisador semântico necessita do analisador sintáctico.

Análise do Discurso

O significado de uma frase individual, inserida num conjunto de frases, pode depender do significado da frase que a antecede e, por sua vez, influenciar o significado da frase que a segue. Com isto, existe a necessidade de que todo o contexto de um conjunto de frases seja analisado, para que se possa compreender o sentido individual de cada frase, principalmente quando se trata de textos ou diálogos, tornando necessária a análise de discurso.

Para a frase utilizada no Exemplo 1 "Eu quero imprimir o arquivo.init do Mário", foi já possível determinar o objectivo que se pretende transmitir, tornando necessária a integração no discurso, para especificar quem é o indivíduo citado na frase. Para que isso possa ser esclarecido, é necessário um modelo de contexto que permita descobrir que o utilizador fez o pedido (quem digitou "Eu") e quem é o utilizador "Mário". Sabendo-se, quem é o Mário, pode definir-se qual ou quais os arquivos com extensão ".init", cujo proprietário é o Mário. Este tipo de reconhecimento só será possível, se o

analisador for capaz de compreender diversas frases, percorrendo uma grande base de conhecimento ou se forem introduzidas fortes restrições de domínio[5] do discurso[6], de maneira a diminuir a base de conhecimento (Specia, 2000).

Processamento Pragmático

A interpretação geral de toda a frase analisada é feita através do processamento pragmático.

Esta análise é importante porque uma frase, após ser reconstruída, pode não ter na sua estrutura nenhuma informação de interesse para a interpretação. Como exemplo, considere-se a frase "Bela camisa Joãozinho", em que podem ser identificados dois sentidos distintos, dependendo da interpretação que lhe for dada.

A pragmática tem uma abrangência muito maior do que simplesmente analisar uma frase isolada, uma vez que analisa todo o discurso. Cada frase pode ter significados que poderão gerar sentidos diferentes podendo comprometer totalmente o significado do texto analisado.

Depois de analisadas as fases necessárias para um sistema computacional interpretar uma frase em Linguagem Natural é apresentada uma tabela com um resumo contendo o objectivo de cada fase:

Níveis de Processamento de Linguagem Natural	
Nível	**Abordagem**
Léxico	Definição das palavras e a sua classe.
Sintáctico	Formação e estrutura de frases.
Semântica	Significado associado à estrutura frásica.
Discurso	Contexto.
Pragmático	Contexto Linguístico + Experiências Humanas.

Tabela 1 – Significado/Resumo dos níveis de Processamento de Linguagem Natural (Von Wangenheim, 1993)

[5] Conjunto de palavras usadas num discurso específico.
[6] Texto ou fala, podendo ser dividido em unidades menores denominadas frases.

- **a Modelagem Estatística de Linguagem**

Segundo Kukich (Kukich, 1992), modelos estatísticos de linguagem são tabelas com estimativas de probabilidade condicional para algumas ou todas as palavras numa linguagem, que especificam a probabilidade da existência de uma palavra no contexto de outras palavras.

Através do exemplo apresentado em (Almeida, et al., 2004), um Modelo Estatístico de Linguagem (MEL) de trigrama de palavra determina a probabilidade para a próxima palavra, condicionada pelas duas palavras anteriores; um MEL de bigrama fornece a probabilidade da próxima palavra condicionada pela palavra anterior e um MEL de colocação fornece a probabilidade de certas palavras ocorrerem a uma certa distância de outra palavra linguisticamente relacionada (por exemplo, a cinco posições, seja para a direita ou para a esquerda).

As probabilidades utilizadas nos MELs são provenientes de grandes quantidades de texto, onde o conceito "grandes" pode atingir dezenas de milhares de palavras, exigindo sistemas computacionais com capacidades suficientes para o processamento de tais quantidades de dados. Por exemplo, é possível constatar que o tamanho de uma tabela de probabilidades de um trigrama cresce de forma exponencial relativamente ao tamanho do léxico (dicionário), ou seja, para um modelo trigrama completo com um léxico de apenas 5.000 palavras teríamos 25.000 milhões de entradas.

É notório que a maioria das entradas tem probabilidade zero. Mas de acordo com a Lei de Zipf (Zipf, 1935), a frequência de ocorrência de um termo é quase inversamente proporcional à sua probabilidade, levando a que haja sempre uma grande cauda de termos que ocorrem somente uma vez. Esta questão levanta um problema ao nível da computação de MELs, pois, nestes casos, existe esforço computacional. Para minorar este problema, algumas técnicas eficientes foram desenvolvidas nos últimos anos, por Bahl(Bahll, et al., 1989)(Bahl, et al., 1983) e Brown(Brown, et al., 1990a).

2.2.5. Conclusão

Na identificação de erros em palavras, apesar da análise n-grama poder ser útil para detectar erros gerados por uma máquina, como aqueles produzidos pelo reconhecimento óptico de caracteres (OCR), tem provado ser menos preciso para a detecção de erros gerados por humanos através da linguagem natural (Kukich, 1992). Assim, a maioria das actuais técnicas de correcção ortográfica utiliza a pesquisa em dicionário para a detecção de erros. Sendo o tempo de acesso um factor importante

quando o tamanho do dicionário é moderado a grande (por exemplo, mais do que 20.000 entradas), algoritmos eficientes para a exacta correspondência utilizam, principalmente, as tabelas *hash*. O dicionário deve ser cuidadosamente adaptado ao domínio do discurso, de modo a evitar a defraudar o utilizador com demasiadas respostas falsas e rejeições.

Relativamente à correcção dependente do contexto, consegue-se facilmente concluir que a técnica ideal ainda não existe. Tal sistema teria que ter uma ampla cobertura lexical e ser capaz de corrigir erros ortográficos em palavras curtas, médias e longas em tempo real e com uma precisão quase perfeita na primeira correcção candidata.

A terceira e última problemática refere-se à correcção dependente do contexto. Este tópico envolve a utilização do contexto para detectar e corrigir erros. Estes erros podem resultar no nível morfológico, sintáctico, semântico, do discurso ou processamento pragmático. Por outro lado, podem ser utilizados métodos estatísticos que utilizam modelos como bigramas ou trigramas para detectar e corrigir erros ortográficos.

Após analisar as diferentes problemáticas da correcção automática de erros ortográficos podemos concluir que esta temática ainda tem um longo caminho a percorrer até conseguirmos ter sistemas robustos, eficientes e eficazes de forma a satisfazer as expectativas de quem utiliza e espera respostas em tempo-real.

2.3. Construção automática de respostas (Information retrieval / Question answering)

2.3.1. Introdução

A temática da construção automática de respostas está normalmente associada aos sistemas de pergunta e resposta (*question answering*), uma especialização dos sistemas de recuperação de informação (*information retrieval*). Assim, iremos de seguida apresentar esses conceitos.

Os tradicionais sistemas de *information retrieval* focalizam-se na procura e elaboração de um ranking de documentos em resposta a uma questão colocada por um utilizador. O problema da recuperação de informação surge quando existe grande quantidade de informação para qual é necessário criar um acesso rápido e esse acesso se está a tornar cada vez mais difícil. Isso pode originar que informação relevante deixe de ser utilizada devido à dificuldade de acesso à mesma. Um

sistema de recuperação de informação permite procurar informação através de consultas, normalmente feitas por meio de palavras-chave, que são pesquisadas na base de dados que contém a informação (Rijsbergen, 1979).

Os sistemas de *question answering (QA)*, são uma especialização dos sistemas de recuperação de informação, que recebem uma pergunta de âmbito mais específico, normalmente em linguagem natural, e apresentam uma resposta directa normalmente acompanhada por uma referência relativa à fonte de onde foi extraída a resposta (Seo, 2002).

No desenvolvimento do sistema proposto iremos construir de forma automática as respostas a serem enviadas aos remetentes recorrendo a algumas técnicas dos sistemas *question answering* que serão apresentadas nos capítulos seguintes.

2.3.2. Question Answering

Para que um sistema *question answering* funcione de acordo com o conceito acima descrito foi desenvolvido um guia (Hirschman, 2001)(Burger, et al., 2002), sendo identificadas cinco características fundamentais para este tipo de sistemas:

1. As respostas devem ser dadas em tempo real, independentemente da complexidade da questão, da dimensão e multiplicidade das fontes de dados ou da ambiguidade da pergunta;
2. Respostas imprecisas ou incorrectas são soluções piores do que não haver resposta. É expectável que o sistema consiga lidar com contradições. Não haverá uma resposta se esta não puder ser extraída das fontes de informação do sistema;
3. Deve-se ter em conta a usabilidade do sistema, em que o conhecimento do sistema deve satisfazer as necessidades do utilizador;
4. Respostas distribuídas por várias fontes que exigem técnicas de fusão e que combinam respostas parciais a partir de diferentes fontes devem ser coerentes;
5. A resposta deve ser relevante dentro de um contexto específico/tarefa. O contexto pode ser utilizado para esclarecer a questão e resolver ambiguidades. O sistema de avaliação deve ser centrado no utilizador.

Após uma análise do guia tentaremos respeitá-lo ao máximo, de forma a desenvolver um sistema robusto e que esteja de acordo com as expectativas de quem utiliza um sistema deste género, mas sempre cientes de que poderá não ser possível cumprir todos os pontos apresentados, sobretudo devido aos problemas intrínsecos dos sistemas *question answering*.

Um dos problemas a ter em conta nos sistemas de *question answering* consiste na tentativa de resolver o mapeamento da terminologia utilizada pelo utilizador para a terminologia utilizada pelas fontes de informação, através de medidas similaridade sem afectar a sua portabilidade. Além disso, independentemente do tipo de consulta, qualquer sistema de *question answering* de linguagem natural tem de lidar com a ambiguidade. Consultas feitas directamente a informação semântica ainda constituem uma área de investigação recente. No entanto, os resultados de muitas áreas de investigação como a selecção e hierarquização de ontologias[7] (o quanto satisfazem as consultas do utilizador), desambiguação do sentido da palavra ("laranja", pode ser uma cor ou uma fruta), podem ser aplicados (Lopez, et al., 2007).

2.3.3. Domínio

A investigação de sistemas *question answering* levou a que se distinguissem dois tipos diferentes de sistemas (Magnini, 2002): de domínio aberto e de domínio fechado.

Sistemas de *question answering* de domínio aberto lidam com questões sobre quase tudo e só podem confiar em ontologias gerais (Magnini, 2002) e no conhecimento do mundo. Por outro lado, estes sistemas geralmente dispõem de muito mais dados disponíveis a partir dos quais se pode extrair a resposta. Os sistemas de domínio aberto podem referir-se a situações em que são aceites um tipo ilimitado de questões.

Por outro lado, os sistemas de *question answering* de domínio fechado lidam com questões de âmbito de um domínio específico (por exemplo, medicina ou previsão meteorológica) e geralmente correspondem a uma tarefa mais fácil, uma vez que os sistemas de PLN podem explorar os conhecimentos específicos de domínio, frequentemente formalizados em ontologias.

2.3.4. Desafios

Várias conferências e *workshops* recentes têm incidido sobre aspectos da área de investigação dos sistemas *question answering*. O *Text Retrieval Conference* (TREC), iniciado em 1999, tem patrocinado este tipo de sistemas, avaliando os que respondem a perguntas factuais através da consulta dos documentos fornecidos pelo TREC. Um número significativo de sistemas apresentados no TREC tem combinado com sucesso a recuperação da informação (*information retrieval*) com técnicas de processamento da linguagem natural.

[7] Uma ontologia define as regras que regulam a combinação entre termos e relações num determinado domínio de conhecimento.

Estas conferências, *workshops* e avaliações estão a abrir novos caminhos para investigadores no domínio do *question answering*. A seguir será fornecida uma visão geral de algumas dimensões da pesquisa (Hirschman, 2001):

- **Aplicações** – Os sistemas *question answering* têm muitas aplicações. Podemos subdividir estas aplicações baseadas na fonte das respostas: dados estruturados (bases de dados), dados semi-estruturados (por exemplo, campos para comentários em bases de dados) ou texto livre. Podemos ainda distinguir entre pesquisa ao longo de um conjunto de colecções pré-definidas, tal como utilizadas no TREC, pesquisa através da Web, pesquisa sobre uma colecção ou livro, por exemplo, uma enciclopédia e pesquisa num único texto.

- **Utilizadores** – Os utilizadores podem variar entre os que utilizam o sistema pela primeira vez ou os que utilizam de forma frequente ou ainda os utilizadores avançados que poderão utilizar o sistema de forma rotineira para a realização do trabalho do dia-a-dia. Desta forma, diferentes tipos de utilizadores requerem interfaces diferentes, fazem perguntas diferentes e esperam diferentes tipos de resposta. Para os utilizadores que utilizam o sistema pela primeira vez, pode ser importante explicar as limitações do sistema, de modo a que o utilizador possa interpretar correctamente as respostas devolvidas. Para utilizadores experientes, é desejável desenvolver e actualizar um modelo do utilizador, para que as respostas possam enfatizar novas informações e omitir informações anteriormente fornecidas ao utilizador.

- **Tipos de questões** – Quanto às questões, podem ser distinguidas pelo tipo de respostas: factuais, de opinião ou sumárias. Está comprovado que alguns tipos de perguntas são mais difíceis do que outros. Por exemplo, questões do tipo *como* e *porquê* são normalmente mais difíceis porque requerem que se perceba a origem e as suas relações e estas são normalmente expressos como cláusulas ou frases separadas (Hirschman, 1999). Assim, se um sistema consegue analisar e prever o tipo de resposta esperado, então o espectro de respostas possíveis é significativamente reduzido. Determinados tipos de questões são mais difíceis de responder devido à dificuldade em reduzir o espectro de respostas.

- **Tipos de respostas** – As respostas podem ser longas ou curtas, podem ser listas ou texto. Podem variar com a utilização pretendida e com os destinatários. Por exemplo, se um utilizador deseja uma justificação, isso requer uma resposta maior. Mas geralmente a compreensão de um simples texto exige respostas curtas (frases). Existem também diferentes metodologias para a construção de uma resposta: através da extracção (cortando e colando pequenos trechos do(s) documento(s) original(s) contendo a resposta) ou através

da sua geração. Se a resposta é extraída a partir de múltiplas frases ou vários documentos, a coerência de uma resposta devolvida pode ser reduzida, exigindo a geração para sintetizar as peças recolhidas para formar uma resposta coerente.

- **Avaliação** – O que faz uma boa resposta? Uma boa resposta é longa, contendo contexto suficiente para justificar a sua escolha como uma resposta? O contexto é útil se o sistema apresenta várias respostas candidatas, porque permite ao utilizador escolher uma resposta correcta, mesmo quando a resposta escolhida pelo utilizador não é a melhor para o sistema. No entanto, existem casos em que as respostas curtas podem ser uma melhor solução. As experiências dos TREC, relativamente à avaliação do *question answering*, demonstram que é mais fácil devolver segmentos mais longos que contêm uma resposta embebida do que segmentos curtos.

- **Apresentação** – Em situações reais de procura de informações, existe um utilizador que interage com um sistema em tempo real. O utilizador regularmente começa com uma questão geral e o sistema fornece *feedback* pela devolução de muitos documentos. O utilizador, em seguida, reduz o âmbito da procura, iniciando uma espécie de diálogo com o sistema. Facilitar esse diálogo, provavelmente irá aumentar a satisfação do utilizador e a usabilidade. Além disso, se as interfaces fossem capazes de lidar com o reconhecimento de fala e com o diálogo escrito, os sistemas *question answering* poderiam proporcionar o acesso a informações disponibilizadas na Web através de uma simples conversa (uma área de grande interesse comercial, sobretudo para as telecomunicações e fornecedores de conteúdos Web). Até à data, tem havido pouco trabalho desenvolvido sobre interfaces para *question answering*. Têm havido poucas avaliações sistemáticas para encontrar qual a melhor forma de apresentar as informações ao utilizador, quantas respostas devem ser apresentadas ao utilizador, a quantidade de contexto a fornecer, ou se se deve fornecer respostas completas ou respostas curtas com um resumo anexado ou apontadores, etc. Esta é uma área que irá receber maior atenção quando as interfaces dos sistemas de *question answering* começarem a ser implantadas em situações comerciais com o necessário retorno económico.

2.3.5. Conclusão

Apesar dos 40 anos de actividade, a área de investigação em *question answering* está apenas no seu início. A atracção pelo desafio de desenvolver sistemas *question answering* é visível (a grande evolução do TREC é um reflexo dessa atracção) e altamente relevante: mesmo soluções limitadas têm

trazido mais-valias para a recuperação de documentos (*document retrieval*). É também estimulante verificar a troca de ideias entre investigadores de processamento da linguagem natural, recuperação de informação e inteligência artificial (Hirschman, 2001).

É esta troca de ideias, divulgada em documentos científicos, que iremos tentar transpor para o desenvolvimento do nosso sistema. Para a análise e resposta a uma mensagem de correio electrónico será deveras importante uma sólida integração entre processamento de linguagem natural e a recuperação de informação, mais especificamente *question answering.*

As conclusões resultantes da recente evolução científica, consequência também do dinamismo do TREC, têm apresentado bons resultados de abstracções dos problemas reais. Apesar de excelentes guias para o nosso desenvolvimento, talvez aqui surjam alguns problemas na implementação de um problema bem real, que é a resposta de forma automática e baseada no contexto a mensagens de correio electrónico, que nos propomos resolver para a Divisão Académica do ISEP.

2.4. Algoritmos de Classificação

2.4.1. Introdução ao *Naïve Bayes*

Olhando um pouco para a história, verificamos que até ao início do século XVIII, os problemas relacionados com a probabilidade de certos eventos, dadas certas condições, eram resolvidos praticamente como hoje os resolvemos. Por exemplo: "dado um número específico de bolas negras e brancas numa urna, qual é a probabilidade de sortear uma bola preta?" Este tipo de problemas são denominados de *forward probability*. Porém, o problema inverso começou a despertar o interesse dos matemáticos da época: "dado que uma ou mais bolas foram sorteadas, o que pode ser dito sobre o número de bolas brancas e pretas na urna?"(Oguri, 2006).

Foi então que um ministro inglês do século XVIII, Thomas Bayes, desenvolveu uma teoria para a resolução de tais problemas (Bayes, 1763), considerada pelo mundo científico como revolucionária. É precisamente este tipo de raciocínio que necessitamos para aplicar um classificador.

O classificador *Naïve Bayes* é um dos mais eficientes e eficazes algoritmos de aprendizagem indutiva para *machine learning*[8] e *data mining*[9]. Este classificador assume que existe independência

[8] É um ramo da inteligência artificial dedicado ao desenvolvimento de algoritmos e técnicas que permitam ao computador aprender através de dados (ex. base de dados).

entre os termos de um documento. Esta visão simplista é, por vezes, criticada por não representar a realidade. Porém, Domingos e Pazzani (Domingos, et al., 1997) mostraram, através de um trabalho teórico, que a suposição de independência de palavras na maioria dos casos não prejudica a eficiência do classificador.

O classificador é denominado ingénuo (*naïve*) por assumir que os atributos são condicionalmente independentes, ou seja, a informação de um evento não é informativa sobre nenhum outro. Apesar de ser denominado de "ingénuo" e simplista, este classificador apresenta dos melhores desempenhos em várias tarefas de classificação. Esse desempenho pode ser consultado, por exemplo, em (Chakrabarti, 2002) e (McCallum, et al., 1998).

2.4.2. Fundamentos teóricos

Neste subcapítulo serão apresentados os fundamentos sobre probabilidade condicional na qual assenta o classificador *Naïve Bayes*. Define-se a probabilidade condicional de um evento A, tendo ocorrido um evento B (com probabilidade diferente de zero), como sendo mostrado pela Equação 1:

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

Onde $P(A \cap B)$ é a probabilidade de ocorrência simultânea dos eventos A e B, isto é, probabilidade conjunta de A e B (também descrita simplesmente como $P(A, B)$).

Para $P(A) \neq 0$, pode-se escrever também (Equação 2):

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

Combinando as Equações 1 e 2 tem-se a principal forma do teorema de Bayes mostrado na Equação 3:

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

Esta equação define o teorema de Bayes onde, P(B) é a probabilidade *a priori* de B. P(B|A) é a probabilidade *a posteriori* do evento B dado a ocorrência de A. P(A|B) é chamada de probabilidade condicional do evento A dada a ocorrência do evento B, quando conhecemos B e não conhecemos A.

[9] É o processo de explorar grandes quantidades de dados à procura de padrões consistentes, como regras de associação ou sequências temporais, para detectar relacionamentos sistemáticos entre variáveis, detectando assim novos subconjuntos de dados.

Se conhecemos A mas não conhecemos B, então é chamada de verosimilhança. A verosimilhança ou *likelihood* de P(A|B) é escrita como L(B|A). P(A) é a probabilidade *a priori* de A e age como uma constante de marginalização fazendo com que o resultado fique sempre entre o intervalo 0 e 1. Com essas definições podemos escrever a regra de Bayes como:

$$prob.posteriori = \frac{\text{verosimilhança} \times \text{prob. priori}}{\text{constante} - \text{de} - \text{normalização}}$$

A probabilidade *a priori* P(B) é a probabilidade marginal do evento B levando-se em conta somente as informações conhecidas sobre ele, ou seja, é a distribuição probabilística que expressa a incerteza sobre B antes que dados relevantes sejam levados em consideração. Frequentemente a probabilidade *a priori* é a pura estimação subjectiva de um especialista.

A verosimilhança ou *likelihood* é o inverso da probabilidade condicional. Dado o evento B, usamos a probabilidade condicional P(A|B) para concluir sobre o evento A. Mas, se o evento A é fornecido, usamos a função de verosimilhança L(B|A) para concluir sobre o evento B. A função de verosimilhança é uma função de probabilidade condicional considerada em função do seu segundo argumento, mantendo-se o primeiro fixo.

2.4.3. Classificador *Naïve Bayes*

O classificador *Naïve Bayes* pode ser visto como um tipo especial de modelo gráfico probabilístico, sendo considerado como um grafo acíclico direccionado (DAG – *Directed Acyclic Graph*). Nesse modelo as variáveis são representados pelos nós e os arcos representam a existência de influências directas entre as variáveis conectadas (Figura 5). A intensidade dessas influências é expressa pelas probabilidades condicionais (Pearl, 1988).

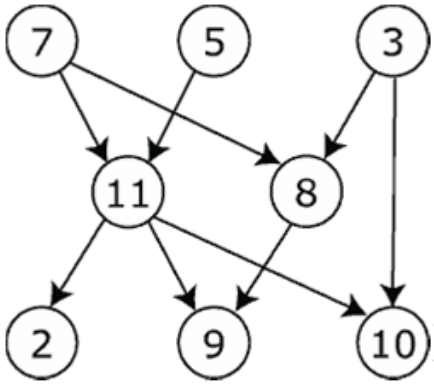

Figura 5 – Exemplo de um gráfico acíclico direccionado (DAG) (Maia, 2005).

Desta forma, o classificador *Naïve Bayes* pode ser visto como uma rede Bayesiana com uma estrutura em estrela (Figura 6) (Borgelt, et al., 2002).

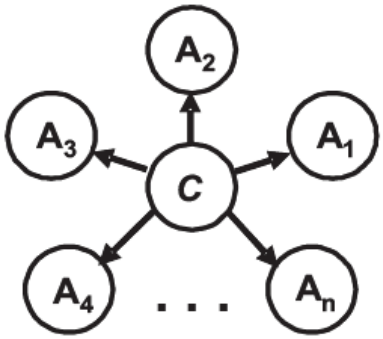

Figura 6 – Estrutura em estrela do Classificador *Naïve Bayes* (Maia, 2005).

A rede Bayesiana é uma das ferramentas que lida com informações incertas baseada na teoria das probabilidades. É a representação de uma distribuição de probabilidade conjunta sobre todas as variáveis representadas pelos nós do grafo. Se $X_{(1)},\ldots,X_{(n)}$ são variáveis e pai(A) é o nó pai de A, então a distribuição conjunta de $X_{(1)}$ até $X_{(n)}$ é representado pelo produto das distribuições das probabilidades $p(X_i \mid pai(X_i))$ para i = 1 até n. Se o nó X não possuir pai, então ele é dito um nó incondicional(Maia, 2005).

O classificador *Naïve Bayes*, devido à sua estrutura simplificada, apresenta inúmeras vantagens. A sua inferência, no nosso caso a classificação, é alcançada em tempo linear. A estrutura do *Naïve Bayes* pode ser facilmente actualizada através da adição ou remoção de evidências no modelo. A suposta independência condicional entre as variáveis nem sempre é verificada podendo trazer influências negativas ao resultado da inferência(Maia, 2005).

2.4.4. Conclusão

O classificador *Naïve Bayes* é simples, rápido e de fácil implementação. Apesar de utilizar a argumentação da existência de independência probabilística, o que em muitos casos não se verifica, tem sido amplamente utilizado em diversas áreas de pesquisa e com muito sucesso. Ao longo dos tempos têm surgido propostas de variações ao seu modelo, observando-se melhorias nos resultados da classificação, mas trazendo prejuízo para o desempenho do sistema que suporta o classificador, dado que existe um aumento no consumo dos recursos do mesmo. É de salientar o trabalho de Friedman e Goldszmidt (Friedman, et al., 1996) que propuseram a interligação entre as variáveis, quebrando o paradigma da independência condicional e trazendo maior liberdade para o modelo; o trabalho apresentado por Borgelt e Gebhardt (Borgelt, et al., 1999), com um classificador probabilístico que é muito mais simples de ser implementado, mas não consegue lidar com variáveis contínuas; e por

último, o trabalho de Langley e Sage (Langley, et al., 1994) que apresentaram uma simplificação do modelo *Naïve Bayes* (Maia, 2005).

3. Projecto SiRAC

3.1. Introdução

A Divisão Académica poderá ser considerada como um *Contact Center*, pois tem como objectivo ser a ponte entre os alunos e a burocracia inerente ao processo educativo, sendo a mensagem de correio electrónico a forma privilegiada de contacto. Desta forma é natural haver a recepção semanal de centenas de mensagens, o que acarreta a necessidade de existirem recursos humanos dedicados a esta tarefa. Estes recursos humanos lêem a mensagem e verificam em que tipo(s) de questão(s) se enquadra, e utilizam respostas pré-definidas para a construção da resposta.

É neste contexto que surge o projecto SiRAC que tem como objectivo responder ao maior número possível de mensagens recepcionados de forma automática, libertando dessa forma os recursos humanos para outras tarefas. Para o desenvolvimento de um sistema deste género é necessário ter em consideração que as mensagens não têm uma estrutura rígida, pois estão em linguagem natural, exigindo processamento para a identificação das questões, através da extracção de palavras e expressões relevantes. A resposta é construída através de *templates*, de acordo com o domínio da questão.

3.1.1. Definição do Problema

No desenvolvimento deste projecto foi utilizado um conjunto de mensagens recepcionadas na Divisão Académica, bem como as respectivas respostas. De salientar que as respostas foram construídas a partir de um conjunto de *templates* disponíveis a quem procede ao envio da resposta às mensagens.

Uma simples mensagem, recepcionado na Divisão Académica, pode conter várias questões, requerendo a utilização de múltiplos *templates* para a construção da resposta. O principal problema reside na necessidade do sistema conseguir determinar as associações entre as questões e os *templates*.

Consideremos $\{Q_1,\ldots,Q_m\}$ as *queries* das mensagens e $\{R_1,\ldots,R_m\}$ as respostas correspondentes e que a *query* Q_i contempla as questões $\{q_1,\ldots,q_m\}$ que são mapeadas nos *templates* $\{t_1,\ldots,t_m\}$ para construir a resposta R_i.

O problema pode pois ser definido da seguinte forma: dada uma mensagem Q_i, é necessário encontrar o conjunto de questões q_s presentes e depois fazer o mapeamento nos *templates* correspondentes, t_s, que são utilizados na composição da resposta R_i.

Baseando-nos nas associações definidas podemo-nos focalizar no problema de construção automática de respostas para as *queries*. Quando chega uma nova *query* ao sistema será necessário identificar as questões contidas e depois construir uma resposta, utilizando os mapeamentos feitos anteriormente.

De salientar que o objectivo deste projecto não é o de responder a todos as mensagens recepcionados na Divisão Académica, o que se poderia tornar extremamente complexo devido à (praticamente) infinita tipologia das questões colocadas através deste meio de comunicação. Para este projecto partiu-se com o objectivo de tentar responder ao maior número possível de questões colocadas, no âmbito dos *templates*/respostas-tipo definidos *a priori*. Estes *templates* foram criados devido à repetição em inúmeras mensagens de questões semelhantes pressupondo respostas semelhantes e foram considerandos, no âmbito deste projecto, os seguintes:

- Tipo 1 – Horário de funcionamento/atendimento da Divisão Académica;
- Tipo 2 – Acesso aos cursos do ISEP para alunos que já possuem uma licenciatura;
- Tipo 5 – Acesso aos cursos do ISEP para alunos com mais de 23 anos (Questões relacionadas com o curso preparatório);
- Tipo 6 – Informações sobre o acesso ao Ano Zero;

- Tipo 7 – Acesso aos cursos do ISEP através de Reingresso;
- Tipo 8 – Acesso aos cursos do ISEP através de transferência de outro curso superior;
- Tipo 10 – Acesso aos cursos do ISEP para alunos que possuem um CET;
- Tipo 11 – Informações dos cursos do ISEP relacionadas com a Ordem dos Engenheiros.

A análise ao sistema a que nos propomos, permitiu identificar as fases de desenvolvimento que são apresentadas na Figura **7** e que serão detalhadas nos subcapítulos seguintes.

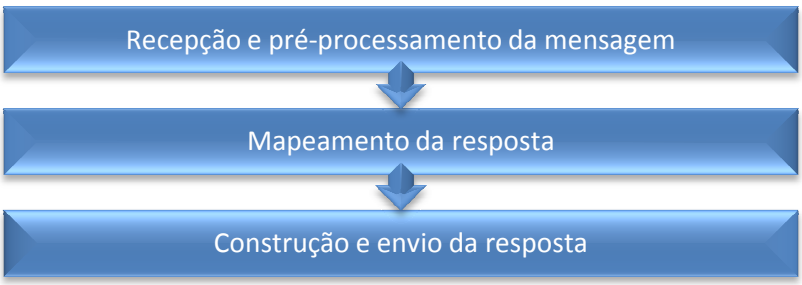

Figura 7 – Fases do projecto

3.2. Recepção e pré-processamento da mensagem

O pré-processamento de textos consiste na transformação de documentos textuais num formato estruturado, tal como uma tabela atributo-valor, para que possam ser aplicados algoritmos de classificação para a extracção de conhecimento dessa informação textual. Porém, essa transformação é um processo pesado e demorado que deve ser feito com cuidado para que o conhecimento extraído seja relevante e útil.

Uma das dificuldades desta fase é que as mensagens não estão em formato estruturado. Esse facto traduz-se numa limitação à utilização de algoritmos de classificação, pois esses algoritmos geralmente necessitam que os dados estejam representados de forma estruturada. A transformação de textos não estruturados em dados estruturados requer pré-processamento.

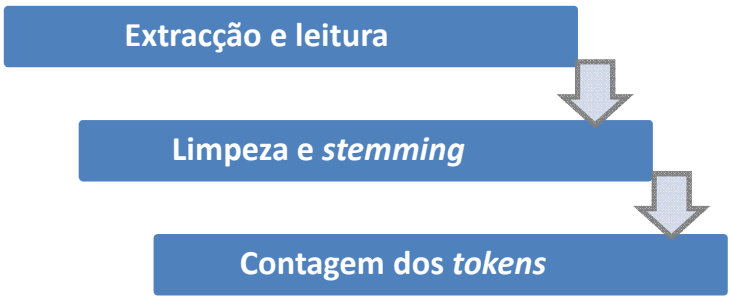

Figura 8 – Etapas do Pré-processamento

O pré-processamento da mensagem é a primeira fase no desenvolvimento do sistema e comporta diversas etapas (Figura 8) com o objectivo de preparar a mensagem para ser processada. Desta forma é necessário fazer uma leitura correcta da mensagem, limpando elementos sem valor semântico para uma posterior radicalização das palavras (os termos derivados de um mesmo radical serão vistos como o mesmo termo, minimizando assim a perda de informação). Por último será feita a contagem dos tokens de acordo com determinados pressupostos.

3.2.1. Extracção e leitura

O primeiro passo consiste na extracção do corpo da mensagem, que será processado pelo sistema. Sabendo que as mensagens de correio electrónico são compostas tanto por texto simples como por código HTML, torna-se necessário proceder à remoção de todas as *tags*[10] de HTML existentes, uma vez que não fazem parte da mensagem a considerar para processamento.

Após esta remoção resta apenas o texto relevante, que terá que ser particionado em *tokens*. A *tokenização* é utilizada para decompor a mensagem em cada termo que a compõe. Esse processo é trivial para uma pessoa que tenha conhecimento da estrutura da linguagem, mas para um programa de computador pode ser mais complicado. No nosso caso, os delimitadores utilizados são: espaço em branco entre os termos; quebras de linhas; tabulações e sinais de pontuação (ponto final, vírgula…). Tanto os grupos de caracteres como os sinais de pontuação tornam-se *tokens* na nova sequência. Os restantes são descartados. Podemos constatar isso através do exemplo a seguir apresentado:

Exemplo 3 – Corpo da mensagem
Boa tarde, Venho por meio deste solicitar informações sobre o acesso ao ensino superior Maiores de 23. Gostaria de saber, se possível, qual a data para as inscrições e qual o endereço do website onde poderei fazer a candidatura electrónica. Sem mais assunto, ##### #####[11]
Corpo da mensagem depois da *tokenização*
'Boa','tarde',',','Venho','por','meio','deste','solicitar', 'informações','sobre','o','acesso','ao','ensino','superior', 'Maiores','de','23','.','Gostaria','de','saber',',','se', 'possível',',','qual','a','data','para','as','inscrições', 'e','qual','o','endereço','do','website','onde','poderei', 'fazer','a','candidatura','electrónica','.','Sem','mais', 'assunto',',','#####','#####'

Tabela 2 – Corpo da mensagem antes e após a *tokenização*.

[10] São estruturas de linguagem de marcação que consistem em breves instruções, tendo uma marca de início e outra de fim. Há uma tendência para se usar as *tags* apenas como delimitadores de estilo e/ou conteúdo, tanto em HTML quanto em XML.
[11] Por questões de privacidade os nomes serão substituídos por cardinais (#) ao longo desta dissertação.

No próximo passo procede-se à remoção da pontuação, seguindo-se a correcção de grafias incorrectas dos *tokens*, através de uma análise morfológica utilizando o módulo *Jspell*.

O *Jspell* é uma ferramenta *open source* para análise morfológica e corrector ortográfico, originalmente desenvolvido para sistemas baseados em Unix. Está orientado para a verificação ortográfica/morfológica de textos/palavras da língua portuguesa. O *Jspell* foi desenvolvido em 1994 por Ulisses Pinto e José João Almeida[12], sendo actualizado periodicamente.

Esta análise permite verificar se a grafia do *token* é coincidente com a duma palavra encontrada no seu dicionário. Caso tal não se verifique, o módulo apresenta sugestões (como mostra a Tabela 3) utilizando a técnica de distância mínima de edição de 1 e 2 alterações (edições) sobre o *token*.

Sugestão apresentada para o *token* "ensinno"	

`'CAT' => 'nc'`
`'guess' => 'ensino'`
`'N' => 's'`
`'unknown' => 1`
`'G' => 'm'`
	➔ `Nome comum`
	➔ `Sugestão`
	➔ `Singular`
	➔ `Edições necessárias para correcção`
	➔ `Masculino`

Tabela 3 – Informações sobre um *token*.

Assim, para o *token* "ensinno" o sistema sugere a palavra mais próxima (`'guess' => 'ensino'`) e apenas com uma edição (`'unknown' => 1`).

Se for sugerida apenas uma alternativa, o sistema assume que esse é o *token* que inicialmente o emissor da mensagem tenha tido intenção de escrever. Caso sejam apresentadas mais que uma alternativa, o sistema irá fazer uma análise sintáctica, através de regras definidas *a priori* no sistema. Terá que haver a concordância entre a categoria do *token* alvo de análise com o *token* anterior e/ou posterior, de forma a fazer a selecção do *token* mais apropriado para ser assumido como correcto. A implementação da análise sintáctica foi baseada numa técnica de correcção dependente de contexto, o MEL de Colocação (apresentado no capítulo "Estado da Arte" – 2.2.4 Correcção dependente de contexto).

3.2.2. Limpeza e *stemming*

De seguida procede-se à limpeza das *stop-words* que podem ser definidas como palavras que do ponto de vista não linguístico não contêm informação e desempenham apenas um papel funcional. São palavras que dependem da linguagem natural utilizada, isto é do idioma em que o texto está escrito. Por exemplo para a língua portuguesa temos: a; à; e; ou; com; como; etc. Ou seja, essa lista é composta por preposições, artigos, advérbios, números e pronomes. A identificação e remoção desta

[12] Pode ser consultado em: http://natura.di.uminho.pt/webjspell/jsolhelp.pl

classe de palavras reduz de forma considerável o tamanho final do léxico, tendo como consequência o aumento de desempenho do sistema.

Em seguida procede-se ao *stemming*, que é um método amplamente utilizado e difundido para diminuir a quantidade de *tokens* necessários para representar uma mensagem, através da transformação de cada *token* para o radical que o originou, por meio de algoritmos de *stemming*. Basicamente, os algoritmos de *stemming* consistem na normalização linguística, na qual as formas variantes de um termo são reduzidas a uma forma comum denominada *stem*. A consequência da aplicação de algoritmos de *stemming* é a remoção de desinências, afixos e vogais temáticas. Com a sua utilização, os termos derivados de um mesmo radical serão vistos como o mesmo termo, ou seja, não se perde detalhe nem precisão.

Corpo da mensagem
'Boa','tarde','Venho','por','meio','deste','solicitar', 'informações','sobre','o','acesso','ao','ensino','superior', 'Maiores','de','23','Gostaria','de','saber','se', 'possível','qual','a','data','para','as','inscrições', 'e','qual','o','endereço','do','website','onde','poderei', 'fazer','a','candidatura','electrónica','Sem','mais', 'assunto','#####','#####'
Corpo da mensagem depois da remoção das *stop-words*
'Boa','tarde','Venho','por','meio','deste','solicitar', 'informações','sobre','acesso','ensino','superior', 'Maiores','23','Gostaria','saber', 'possível','qual','data','para','inscrições', 'qual','endereço','website','onde','poderei', 'fazer','candidatura','electrónica','Sem','mais', 'assunto','#####','#####'
Corpo da mensagem depois do *stemming*
'Boa','tarde','vir','por','meio','deste','solicitar', 'informar','sobre','acesso','ensino','superior', 'maior','23','gostar','saber', 'possível','qual','data','para','inscrição', 'qual','endereço','website','onde','poder', 'fazer','candidatura','electrónica','sem','mais', 'assunto','#####','#####'

Tabela 4 – Corpo da mensagem após remoção de *stop-words* e após *stemming*.

3.2.3. Contagem dos *tokens*

Depois de concluídas as fases anteriores fica-se perante uma lista de *tokens* representativos da mensagem, que servirão para fazer a contabilização de *tokens* (palavras) e de conjunto de *tokens* (expressões) que fazem parte de uma matriz pré-criada (Tabela 5). Essa matriz contem palavras e expressões normalmente existentes no tipo de mensagens previamente definido, de acordo com o tipo de questão (também definido anteriormente).

Tipo de Questão	*Tokens* (palavras)	Conjunto de *tokens* (expressões)
Tipo 1	"secretaria" "funcionamento"	["horário","funcionar"]
Tipo 2	"licenciar"	["possuir","um","licenciatura"] ["possuir","licenciatura"] ["ter","um","curso","superior"] ["ser","licenciar"] ["acabar","licenciatura"]
Tipo 5	"protocolo" "maior" ">" "23" "idade" "calendario" "matriz" "m23" "M23" "preparação" "preparatório"	["exame","matemática"] ["prova","matemática"] ["curso","preparatório"] ["curso","preparar"] ["cursar","preparar"] ["concurso","habilitar","especial"] ["m23"] ["M23"] ["maior","23"] [">","23"] ["+","23"] ["mais","23"] ["exame","23"]
Tipo 6	"zero"	["ano","zero"]
Tipo 7	"reingresso" "ingressar" "reingressar" "regressar"	["equivalência","licenciatura"]
Tipo 8	"transferência" "transferir" "mudança" "mudar" "tranferencia"	["data","transferencia"] ["data","transferência"] ["mudança","estabelecer"] ["mudança","faculdade"] ["mudança","curso"] ["transferência","curso"] ["efectuar","transferência"] ["pedir","tranferência"] ["trocar","curso"] ["transferir","para","ICEP"] ["acabar","licenciatura"] ["frequentar","universidade"]
Tipo 10	"det" "cet" "nível"	["nível","IV"] ["ter","um","cet"] ["ter","um","det"] ["detentor","det"] ["detentor","cet"]
Tipo 11	"ordem"	["ordem","engenheiro"]

Tabela 5 – Palavras e expressões normalmente existentes em mensagens, de acordo com o tipo de questão.

De salientar que a matriz (Tabela 5) também inclui *tokens* que, mesmo após a aplicação do módulo de análise morfológica/corrector ortográfico *Jspell* (apenas permite até 2 edições), e da aplicação da análise sintáctica, ainda não se encontram escritos de forma correcta. Temos como exemplo o *token* "transferência", que ao longo das mensagens da amostra foi possível observar a existência de diversas grafias para esta palavra devido a abreviaturas, falta de acento e/ou erros ortográficos. Desta forma foi necessário fazer a inclusão de *tokens* que após as várias análises continuavam sem a grafia correcta mas que eram essenciais para identificar os tipos de questões presentes. Relativamente ao *token* "transferência" foi adicionado o *token* "tranferencia" à matriz representada através da Tabela 5.

Segundo a Tabela 5, para o Tipo 1 serão contabilizadas todas as ocorrências do *token* `"secretaria"` e `"funcionamento"` pois são palavras normalmente utilizadas em mensagens deste Tipo. As ocorrências da expressão `["horário","funcionar"]`, ou seja, o *token* `“funcionar”` a seguir ao *token* `“horário”` também serão contabilizadas por estarem geralmente representadas em mensagens deste Tipo. De salientar que os *tokens* nesta fase já foram pré-processados, não tendo a sua grafia inicial, que neste caso poderia ser “horário de funcionamento”.

Tipo de Questão	***Tokens* (palavras)**		**Conjunto de *tokens* (expressões)**	
Tipo 1	"secretaria" "funcionamento"	0 0	["horário","funcionar"]	0
Tipo 2	"licenciar"	0	["possuir","um","licenciatura"] ["possuir","licenciatura"] ["ter","um","curso","superior"] ["ser","licenciar"] ["acabar","licenciatura"]	0 0 0 0 0
Tipo 5	"protocolo" **"maior"** ">" **"23"** "idade" "calendario" "matriz" "m23" "M23" "preparação" "preparatório"	0 **1** 0 **1** 0 0 0 0 0 0 0	["exame","matemática"] ["prova","matemática"] ["curso","preparatório"] ["curso","preparar"] ["cursar","preparar"] ["concurso","habilitar","especial"] ["m23"] ["M23"] **["maior","23"]** [">","23"] ["+","23"] ["mais","23"] ["exame","23"]	0 0 0 0 0 0 0 0 **2** 0 0 0 0
Tipo 6	"zero"	0	["ano","zero"]	0
Tipo 7	"reingresso" "ingressar" "reingressar" "regressar"	0 0 0 0	["equivalência","licenciatura"]	0
Tipo 8	"transferência" "transferir" "mudança" "mudar" "tranferencia"	0 0 0 0 0	["data","transferencia"] ["data","transferência"] ["mudança","estabelecer"] ["mudança","faculdade"] ["mudança","curso"] ["transferência","curso"] ["efectuar","transferência"] ["pedir","tranferência"] ["trocar","curso"] ["transferir","para","ICEP"] ["acabar","licenciatura"] ["frequentar","universidade"]	0 0 0 0 0 0 0 0 0 0 0 0
Tipo 10	"det" "cet" "nível"	0 0 0	["nível","IV"] ["ter","um","cet"] ["ter","um","det"] ["detentor","det"] ["detentor","cet"]	0 0 0 0 0
Tipo 11	"ordem"	0	["ordem","engenheiro"]	0

Tabela 6 – Contagem das palavras e expressões existentes na mensagem, de acordo com o tipo de questão.

A cada *token* (palavra) e a cada conjunto de *tokens* (expressão) é atribuído o valor de 1. Assim na Tabela 6 temos representado o somatório de todas as palavras e expressões encontrados por tipo de questão. No final da execução deste módulo teremos o valor atribuído a cada palavra/expressão da mensagem em análise, que é guardada numa nova matriz utilizada no mapeamento da resposta.

3.3. Identificação/Mapeamento da Resposta

A fase que se segue consiste na mineração de dados recolhidos através da contabilização feita anteriormente. Desta forma, a mineração dos dados irá utilizar um conjunto de ferramentas e técnicas que, através do uso de algoritmos de aprendizagem ou classificação, são capazes de explorar um conjunto de dados, extraindo ou ajudando a evidenciar padrões e auxiliando na classificação das respostas.

Assim, o objectivo será identificar a resposta ou as respostas correctas a enviar ao remetente. Pretende-se ainda que respostas identificadas erradamente não sejam, mas sim redireccionadas de forma a serem respondidas pelos recursos humanos da Divisão Académica. Refira-se que, no tipo de questões que são recepcionadas pela Divisão Académica, é fundamental que não sejam fornecidas respostas incorrectas uma vez que as consequências, para os alunos, poderão ser extremamente negativas podendo ir desde a necessidade de pagamento de multas até à exclusão de um concurso. Daí que, como se verá nos capítulos seguintes, se defina como objectivo primordial (talvez ainda mais do que o de fornecer respostas) o de nunca ser enviada uma resposta errada.

3.3.1. Origem dos dados

Os dados utilizados nas experiências foram fornecidos pela Divisão Académica e são constituídos por 258 mensagens recebidas (com 278 questões) com a respectiva resposta enviada. Após a sua classificação foram encontrados 8 tipos/*templates* de temas distintos, conforme a tabela seguinte:

Origem dos dados	
Tipo	**Número de questões**
Tipo 1	3
Tipo 2	21
Tipo 5	**139**
Tipo 6	4
Tipo 7	37
Tipo 8	63
Tipo 10	8
Tipo 11	3
Total	**278**

Tabela 7 – Origem dos dados.

É notório que, neste conjunto de dados, a maior parte das questões colocadas à Divisão Académica encaixam no Tipo 5, ou seja, informações sobre o acesso aos cursos do ISEP para alunos com mais de 23 anos, o que exigiu mais cuidado no tratamento de mensagens deste tipo e por isso apresentado através do Exemplo 3 (Tabela 8). Naturalmente que tal foi fortemente influenciado pelo período do ano em que se procedeu à recolha dos dados, sendo que, para uma amostragem recolhida ao longo de todo o ano, os resultados seriam significativamente diferentes. No entanto, essa constatação em nada invalida os resultados obtidos no presente trabalho.

Exemplo 3 (já utilizado anteriormente) – Corpo da mensagem 1 (E1)
Boa tarde, Venho por meio deste solicitar informações sobre o acesso ao ensino superior Maiores de 23. Gostaria de saber, se possível, qual a data para as inscrições e qual o endereço do website onde poderei fazer a candidatura electrónica. Sem mais assunto, ##### #####

Tabela 8 – Corpo da mensagem 1 (E1).

Como já foi referido anteriormente, o objectivo deste projecto não é responder a todos as mensagens de forma automática. Existirão casos em que será necessária a intervenção da Divisão Académica para fornecer a resposta, pois o sistema não consegue responder devido ao incumprimento de algum dos requisitos. Para demonstrar essa situação será utilizado o Exemplo 4 (Tabela 9).

Exemplo 4 – Corpo da mensagem 2 (E2)
Boa Tarde! Chamo-me ##### #####, Fui aluno do ISEP entre #### e #### com o número #######. Concluí o antigo bacharelato de 4 anos em engenharia electrotécnica. Pretendia no momento obter o grau de licenciado. Agradecia esclarecimentos sobre como ingressar de novo no ISEP para esse fim, quais as equivalências e procedimentos necessários para as o obter, no âmbito do processo de Bolonha. Agradecimentos de ##### #####

Tabela 9 – Corpo da mensagem 2 (E2).

Outro pormenor relevante é a existência de 258 mensagens com 278 questões, ou seja, existem mensagens com mais que uma questão, exigindo que o sistema consiga identificar as questões relevantes. Através do Exemplo 5 (Tabela 10) será apresentado um desses casos que será alvo de análise.

Exemplo 5 – Corpo da mensagem 3 (E3)
Bom Dia, Pretendo candidatar-me ao curso de engenharia informática no v/instituto, no entanto já possuo uma licenciatura em contabilidade e administração pelo ISCAP, visto as minhas notas serem baixas para concurso especial como titular de curso superior, gostaria de saber se é possível concorrer via maiores de 23 anos e simultaneamente pela via normal fazendo os exames de secundário. Ou se terei de escolher um dos concursos apenas. Fico a aguardar uma resposta. Cumprimentos, ##### #####

Tabela 10 – Corpo da mensagem 3 (E3).

Após o pré-processamento aplicado às diferentes mensagens (E1, E2 e E3), é consultada a Tabela 5 (– Palavras e expressões normalmente existentes em mensagens, de acordo com o tipo de questão.) como referência, permitindo obter uma tabela similar à Tabela 6 (– Contagem das palavras e expressões existentes na mensagem, de acordo com o tipo de questão.), cujo resultado é apresentado na Tabela 11 (que, de forma a simplificar a visualização, agrega informação das 3 mensagens seleccionados para análise).

Uma análise à Tabela 11 permite constatar que a categorização aí efectuada utiliza o conceito *bag-of-words*. Na representação *bag-of-words*, cada palavra é apresentada como sendo uma variável por si só e com um dado peso atribuído. Ao contrário do que acontece com a técnica de extracção de informação (*Information Extraction*), o programa não tenta processar a informação existente, fazendo, em vez disso, uma contagem das palavras que aparecem na mensagem. A aproximação *bag-of-words* é bastante utilizada pois é facilmente aplicável a texto proveniente de diversas fontes. Contudo, tem a desvantagem de não ser capaz de capturar a semântica do texto, ou seja, não faz uma distinção entre diferentes contextos pelo que não é capaz de inferir que uma mesma palavra-chave pode carregar significados distintos.

Tipo de Questão	*Tokens* (palavras)	E1	E2	E3	Conjunto de *tokens* (expressões)	E1	E2	E3
Tipo 1	"secretaria"	0	0	0	["horário","funcionar"]	0	0	0
	"funcionamento"	0	0	0				
Tipo 2	**"licenciar"**	0	**1**	0	**["possuir","um","licenciatura"]**	0	0	**1**
					["possuir","licenciatura"]	0	0	0
					["ter","um","curso","superior"]	0	0	0
					["ser","licenciar"]	0	0	0
					["acabar","licenciatura"]	0	0	0
Tipo 5	"protocolo"	0	0	0	["exame","matemática"]	0	0	0
	"maior"	**1**	0	**1**	["prova","matemática"]	0	0	0
	">"	0	0	0	["curso","preparatório"]	0	0	0
	"23"	**1**	0	**1**	["curso","preparar"]	0	0	0
	"idade"	0	0	0	["cursar","preparar"]	0	0	0
	"calendario"	0	0	0	["concurso","habilitar","especial"]	0	0	0
	"matriz"	0	0	0	["m23"]	0	0	0
	"m23"	0	0	0	["M23"]	0	0	0
	"M23"	0	0	0	**["maior","23"]**	**1**	0	**1**
	"preparação"	0	0	0	[">","23"]	0	0	0
	"preparatório"	0	0	0	["+","23"]	0	0	0
					["mais","23"]	0	0	0
					["exame","23"]	0	0	0
Tipo 6	"zero"	0	0	0	["ano","zero"]	0	0	0
Tipo 7	"reingresso"	0	0	0	["equivalência","licenciatura"]	0	0	0
	"ingressar"	0	**1**	0				
	"reingressar"	0	0	0				
	"regressar"	0	0	0				
Tipo 8	"transferência"	0	0	0	["data","transferencia"]	0	0	0
	"transferir"	0	0	0	["data","transferência"]	0	0	0
	"mudança"	0	0	0	["mudança","estabelecer"]	0	0	0
	"mudar"	0	0	0	["mudança","faculdade"]	0	0	0
	"tranferencia"	0	0	0	["mudança","curso"]	0	0	0
					["transferência","curso"]	0	0	0
					["efectuar","transferência"]	0	0	0
					["pedir","tranferência"]	0	0	0
					["trocar","curso"]	0	0	0
					["transferir","para","ICEP"]	0	0	0
					["acabar","licenciatura"]	0	0	0
					["frequentar","universidade"]	0	0	0
Tipo 10	"det"	0	0	0	["nível","IV"]	0	0	0
	"cet"	0	0	0	["ter","um","cet"]	0	0	0
	"nível"	0	0	0	["ter","um","det"]	0	0	0
					["detentor","det"]	0	0	0
					["detentor","cet"]	0	0	0
Tipo 11	"ordem"	0	0	0	["ordem","engenheiro"]	0	0	0

Tabela 11 – Contagem das palavras e expressões dos exemplos.

De seguida serão aplicadas técnicas de classificação para averiguar qual/quais as questões constantes das mensagens. De forma a permitir a avaliação dos resultados, foram previamente determinados os tipos de cada um dos exemplos anteriormente apresentados. Os resultados foram os seguintes:

- a mensagem do exemplo E1 é do Tipo 5
- a mensagem do exemplo E2 é do Tipo 7
- a mensagem do exemplo E3 é simultaneamente dos Tipos 2 e 5.

3.3.2. *Naïve Bayes* 1

O algoritmo *Naïve Bayes* é um algoritmo de classificação que utiliza o teorema de *Bayes* (2.4.1 Introdução ao *Naïve Bayes*), em que as entradas são consideradas independentes entre si. No entanto, na maioria dos problemas práticos, tal não se verifica (daí o nome *naïve* ou ingénuo). Esta suposição simplifica a abordagem do problema de classificação sem comprometer significativamente a precisão do resultado, tornando este algoritmo computacionalmente menos intenso de que outros, sendo útil para gerar rapidamente modelos de mineração de forma a descobrir as relações entre as colunas de entrada e as colunas previsíveis.

Desta forma a utilização deste algoritmo no desenvolvimento deste projecto justifica-se pelo facto de apresentar dos melhores desempenhos em tarefas de classificação (Chakrabarti, 2002)(McCallum, et al., 1998), em que a sua visão simplista (assume que existe independência entre os termos) na maioria dos casos não prejudica a eficiência do classificador (Domingos, et al., 1997) e de não necessitar de muitos recursos computacionais para a sua utilização.

Aproveitando as potencialidades deste algoritmo surgiu a técnica *Naïve Bayes* 1 que é aplicada na classificação de *tokens* (palavras) da amostra de mensagens, conforme os pressupostos apresentados no capítulo do Estado da Arte – 2.4.2 Fundamentos teóricos, obtendo os resultados apresentados na Tabela 12.

Aplicação do *Naïve Bayes* 1	
Mensagens identificados correctamente	**Mensagens identificados incorrectamente**
178	**80**

Tabela 12 – Aplicação do *Naïve Bayes* 1.

Como se pode constatar pela análise da tabela, obtêm-se 178 mensagens identificadas correctamente de um total de 258, o que resulta numa taxa de acerto de aproximadamente de 69%. Este resultado pode ser considerado bom. No entanto existem 80 mensagens em que as questões foram identificadas incorrectamente e consequentemente seriam respondidas 80 mensagens de forma incorrecta. Como um dos pressupostos do SiRAC é não enviar respostas erradas ao remetente, a solução encontrada foi a de criar requisitos mínimos, apresentados na Tabela 13, de forma a eliminar as mensagens identificadas incorrectamente.

Naïve Bayes 1	
Requisitos mínimos	
Resposta simples Ou Resposta múltipla (tipo com maior valor)	O valor da questão terá que ser maior ou igual a 0.94
Resposta múltipla (segundo tipo com maior valor)	O valor da questão terá que ser maior ou igual a 0.74
Resposta múltipla (terceiro tipo com maior valor)	O valor da questão terá que ser maior ou igual a 0.71

Tabela 13 – Requisitos mínimos para o *Naïve Bayes* 1.

Os valores apresentados na tabela anterior resultam de testes feitos à amostra de mensagens existente, com o objectivo de eliminar todas as perguntas identificadas erradamente. Note-se que a metodologia utilizada foi a de determinar o valor mais elevado para o qual não é reportada nenhuma mensagem incorrectamente identificada. Esta metodologia é, obviamente, muito sensível à amostra utilizada. No entanto, como o objectivo é o de comparar diferentes técnicas que são pouco sensíveis às amostras, tal limitação não invalida as conclusões.

Para que o sistema responda correctamente a todos as mensagens da amostra é necessário que uma mensagem com apenas uma questão tenha o valor de 0.94, caso tenha duas questões a segunda terá que ter 0.74 (o valor para a primeira questão mantém-se) e se a mensagem tiver ainda uma terceira questão terá que ter o valor de 0.71 (mantendo-se o valor para as outras questões). Com a utilização destes requisitos existiram alterações nos valores apresentados quanto ao número de mensagens identificadas correctamente segundo esta técnica (Tabela 14).

Aplicação do *Naïve Bayes* 1			
Mensagens identificados correctamente	**Mensagens sem requisitos mínimos**	**Mensagens identificados incorrectamente**	**Total**
2	**256**	**0**	**258**

Tabela 14 – Aplicação do *Naïve Bayes* 1 com requisitos mínimos.

Analisando a Tabela 14 verificamos que com os requisitos mínimos definidos não são identificadas incorrectamente quaisquer mensagens mas comprometemos as mensagens identificadas correctamente (que são apenas 2), pois todas as outras não apresentam os requisitos necessários. Com estes resultados podemos concluir que esta técnica não servirá os propósitos do sistema, pois com uma taxa de acerto de praticamente 0% não se iria conseguir automatizar o envio de respostas a mensagens.

Para uma melhor percepção do funcionamento desta técnica serão apresentados os valores resultantes da aplicação do *Naïve Bayes* 1 aos exemplos E1, E2 e E3:

Resultados do *Naïve Bayes* 1		
E1	**E2**	**E3**
Tipo 1 => 0.25029 Tipo 2 => 0.26725 **Tipo 5 => 0.60131** Tipo 6 => 0.26725 Tipo 7 => 0.22087 Tipo 8 => 0.20807 Tipo 10 => 0.23489 Tipo 11 => 0.26725	Tipo 1 => 0.24762 **Tipo 2 => 0.52881** Tipo 5 => 0.14873 Tipo 6 => 0.26440 Tipo 7 => 0.43703 Tipo 8 => 0.20585 Tipo 10 => 0.23239 Tipo 11 => 0.26440	Tipo 1 => 0.25029 Tipo 2 => 0.26725 **Tipo 5 => 0.60131** Tipo 6 => 0.26725 Tipo 7 => 0.22087 Tipo 8 => 0.20807 Tipo 10 => 0.23489 Tipo 11 => 0.26725

Tabela 15 – Aplicação da técnica *Naïve Bayes* 1.

Podemos constatar na Tabela 15 que, no caso E1, o Tipo 5 salienta-se com o valor mais elevado (0.60131), bem superior aos restantes fazendo o mapeamento de forma correcta. No caso E2 distingue-se o Tipo 2 com o valor mais elevado (0.52881) mas não correspondendo ao Tipo presente na mensagem (Tipo 7). Relativamente ao caso E3, o Tipo 5 tem o maior valor (0.60131) identificando parte da resposta à mensagem (Tipo 2 e Tipo 5).

Apesar desta técnica conseguir identificar correctamente a questão presente no E1 e parte de E3 não apresenta os requisitos mínimos suficientes (Tabela 13), o que inviabiliza a utilização desta técnica na identificação das respostas dos exemplos.

3.3.3. *Naïve Bayes* 2

O *Naïve Bayes* 2 utiliza novamente todas as potencialidades do algoritmo *Naïve Bayes* (2.4.1-Introdução ao *Naïve Bayes*) sendo um dos mais eficientes e eficazes algoritmos de classificação. A diferença relativamente ao *Naïve Bayes* 1 reside no facto de esta técnica ser aplicada tanto a *tokens* (palavras) como a conjuntos de *tokens* consecutivos (expressões). Desta forma o resultado de cada Tipo de questão (Tabela 5) é o somatório dos resultados obtidos para palavras e expressões do mesmo Tipo de questão. Comparativamente ao *Naïve Bayes* 1, esta técnica apresentará valores de classificação superiores pois, para além das palavras, tem também em consideração as expressões frequentemente utilizadas nos diferentes tipos de questão.

Na Tabela 16 são apresentados os resultados da aplicação do *Naïve Bayes* 2:

Aplicação do *Naïve Bayes* 2	
Mensagens identificados correctamente	**Mensagens identificados incorrectamente**
217	**41**

Tabela 16 – Aplicação do *Naïve Bayes* 2.

Analisando a tabela constatamos que 217 mensagens são identificadas correctamente, o que pressupõe uma taxa de acerto de 84%, bem superior ao *Naïve Bayes* 1. Mas, tal como fizemos anteriormente, a existência de mensagens identificadas incorrectamente implica a criação de requisitos que impeçam que respostas erradas sejam enviadas aos remetentes.

Naïve Bayes 2	
Requisitos mínimos	
Resposta simples Ou Resposta múltipla (tipo com maior valor)	O valor da questão terá que ser maior ou igual a 1
Resposta múltipla (segundo tipo com maior valor)	O valor da questão terá que ser maior ou igual a 0.76
Resposta múltipla (terceiro tipo com maior valor)	O valor da questão terá que ser maior ou igual a 0.73

Tabela 17 – Requisitos mínimos para o *Naïve Bayes* 2.

Os valores presentes na Tabela 17 são o resultado de testes efectuados sobre a amostra até estarmos perante a ausência de mensagens identificadas incorrectamente. Assim para mensagens que apenas contenham uma questão, o valor terá que ser maior ou igual a 1; caso tenha duas questões, a segunda terá que ser maior ou igual a 0.76; se a mensagem tiver três questões, a terceira terá que ser maior ou igual a 0.73. De salientar que estes valores resultam da aplicação destes testes a esta amostra. Após aplicação dos requisitos à amostra de mensagens e segundo o *Naïve Bayes* 2 obtém-se a seguinte tabela:

Aplicação do *Naïve Bayes* 2			
Mensagens identificados correctamente	**Mensagens sem requisitos mínimos**	**Mensagens identificados incorrectamente**	**Total**
0	**258**	**0**	**258**

Tabela 18 – Aplicação do *Naïve Bayes* 2 com requisitos mínimos.

Na Tabela 18 podemos constatar que após a aplicação dos requisitos mínimos o sistema não consegue identificar correctamente nenhuma mensagem, pois não apresentam as condições necessárias.

De salientar que antes da aplicação dos requisitos mínimos esta técnica tinha uma taxa de acerto superior à técnica *Naïve Bayes* 1 (também antes da aplicação dos requisitos mínimos), o que poderia pressupor melhores resultados após aplicar os requisitos, mas consegue ter um valor ainda mais baixo, ou seja uma taxa de acerto de 0%.

Desta forma esta técnica não traria nenhuma melhoria relativamente ao *Naïve* Bayes 1, nem ao tradicional envio de resposta a mensagens feito pelos recursos humanos, pois não iria conseguir responder a nenhuma mensagem. Para demonstrar a ineficácia apresentamos a Tabela 19 que resulta da aplicação concreta do *Naïve Bayes* 2 aos exemplos E1, E2 e E3.

Resultados do *Naïve Bayes* 2		
E1	**E2**	**E3**
Tipo 1 => 0.01440 Tipo 2 => 0.03308 **Tipo 5 => 0.99718** Tipo 6 => 0.01482 Tipo 7 => 0.01361 Tipo 8 => 0.04978 Tipo 10 => 0.03227 Tipo 11 => 0.01482	Tipo 1 => 0.10983 **Tipo 2 => 0.75680** Tipo 5 => 0.36209 Tipo 6 => 0.11304 Tipo 7 => 0.20763 Tipo 8 => 0.37957 Tipo 10 => 0.24608 Tipo 11 => 0.11304	Tipo 1 => 0.01440 Tipo 2 => 0.03308 **Tipo 5 => 0.99718** Tipo 6 => 0.01482 Tipo 7 => 0.01361 Tipo 8 => 0.04978 Tipo 10 => 0.03227 Tipo 11 => 0.01482

Tabela 19 – Aplicação da técnica *Naïve Bayes* 2.

Analisando a tabela verificamos que E1 volta a ser identificado correctamente (Tipo 5) com 0.99718. Pelo contrário E2 continua a ser identificado de forma errada, sendo do Tipo 7 apresenta o maior valor para o Tipo 2 (0.75680). Relativamente ao caso E3, o Tipo 5 com o valor 0.99718 identifica parte da resposta (Tipo 2 e Tipo 5). De referir que se verifica um destaque muito maior nos tipos identificados, comparativamente com os restantes, mas em todos os casos não é respeitado o valor necessário para cumprir os requisitos mínimos, tendo todos os casos que ser tratados pelos recursos humanos da Divisão Académica.

3.3.4. *Bag-of-words*

A técnica *Bag-of-words* utiliza um algoritmo de classificação baseado na presença ou na ausência de termos chave, que são independentes entre si. Cada palavra/expressão é representada como sendo uma variável por si só e com um dado peso atribuído. Esse peso será a contagem do número de vezes que a palavra/expressão aparece na mensagem. Esta simplicidade de aplicação resulta numa boa eficiência e eficácia que justificaram a sua escolha para utilização neste projecto.

Porém, as maiores desvantagens do algoritmo *Bag-of-words* são a não utilização de relações semânticas e sintácticas entre os termos e a desconsideração de outras estruturas complexas, como a diferença do verdadeiro significado de um termo quando redigido em contextos diferentes.

Desta forma, cada *token* ou conjunto de *tokens* representa uma variável com um dado peso, que será o número de ocorrências desse *token* ou conjunto de *tokens*. Na Tabela 20 são apresentados os resultados da aplicação do *Bag-of-words* à amostra de mensagens:

Aplicação do *Bag-of-words*	
Mensagens identificados correctamente	**Mensagens identificados incorrectamente**
205	**53**

Tabela 20 – Aplicação do *Bag-of-words*.

Pela análise da Tabela 20 pode constatar-se que temos uma taxa de acerto de aproximadamente de 79% (superior ao *Naïve Bayes* 1 mas inferior ao *Naïve Bayes* 2) mas com a existência de mensagens identificadas incorrectamente que terão que ser eliminadas para não enviar ao remetente da mensagem uma resposta errada. Desta forma foi necessário, tal como anteriormente, definir requisitos mínimos para garantir a inexistência de respostas erradas.

Bag-of-words	
Requisitos mínimos	
Resposta simples Ou Resposta múltipla (tipo com maior valor)	O valor da questão terá que ser maior ou igual a 3
Resposta múltipla (segundo tipo com maior valor)	
Resposta múltipla (terceiro tipo com maior valor)	

Tabela 21 – Requisitos mínimos para o *Bag-of-words*.

Estes resultados foram conseguidos através de testes efectuados sobre a amostra de mensagens, e verificou-se que quando o valor é maior ou igual a 3 garantimos que não existe mensagem identificada incorrectamente. Através da Tabela 22 podemos verificar o resultado da aplicação desta técnica utilizando os requisitos mínimos:

Aplicação do *Bag-of-words*			
Mensagens identificados correctamente	**Mensagens sem requisitos mínimos**	**Mensagens identificados incorrectamente**	**Total**
104	**154**	**0**	**258**

Tabela 22 – Aplicação do *Bag-of-words* com requisitos mínimos.

Analisando a tabela constatamos que esta técnica irá responder a 40% das mensagens de forma correcta, sem a existência de mensagens enviadas ao remetente desenquadradas das suas questões.

Comparativamente com as técnicas anteriores (*Naïve Bayes* 1 e *Naïve Bayes* 2), a *Bag-of-words* consegue uma taxa de acerto (após aplicação dos requisitos) muito superior e com valores aceitáveis que justifiquem a utilização deste tipo de sistema.

Para comprovar a aplicação do *Bag-of-words* foram utilizados os exemplos E1, E2 e E3 em que os resultados são apresentados na seguinte tabela:

Resultados do *Bag-of-words*		
E1	**E2**	**E3**
Tipo 1 => 0 Tipo 2 => 0 **Tipo 5 => 3** Tipo 6 => 0 Tipo 7 => 0 Tipo 8 => 0 Tipo 10 => 0 Tipo 11 => 0	Tipo 1 => 0 **Tipo 2 => 1** Tipo 5 => 0 Tipo 6 => 0 **Tipo 7 => 1** Tipo 8 => 0 Tipo 10 => 0 Tipo 11 => 0	Tipo 1 => 0 **Tipo 2 => 1** **Tipo 5 => 3** Tipo 6 => 0 Tipo 7 => 0 Tipo 8 => 0 Tipo 10 => 0 Tipo 11 => 0

Tabela 23 – Aplicação da técnica *Bag-of-words*.

De acordo com tabela anterior podemos verificar que o E1 foi correctamente identificado com o Tipo 5 (3), cumprindo os requisitos; para o E2 foram destacados os Tipos 2 e 7 (a resposta correcta apenas é do Tipo 7), mas não cumprem os requisitos; e o E3 identifica correctamente o Tipos 2 (1) e o Tipo 5 (3), mas apenas o Tipo 5 cumpre com os requisitos mínimos. Os resultados da Tabela 23 que estão destacados com sombreado cinzento estão correctamente identificados e cumprem os requisitos necessários.

3.3.5. *Naïve Bayes* 2 + *NT* 1 (*Bag-of-words* com pesos diferentes)

Apesar da técnica *Bag-of-words* apresentar resultados aceitáveis, estão ainda longe de permitir responder automaticamente à totalidade da amostra. Uma possibilidade para a obtenção de melhores resultados consiste na conjugação da aplicação de algumas das técnicas anteriormente apresentadas, Assim decidimos testar a conjugação da *Bag-of-words* (que identifica palavras e expressões) com o *Naïve Bayes* 2 (que também identifica palavras e expressões) de forma a determinar se identificam as mesmas questões na mensagem, dando-lhes um destaque ainda maior e consequentemente melhorar os resultados. A conjugação que se propõe consiste na soma dos valores resultantes de ambas as técnicas. Dessa aplicação resultou a seguinte tabela:

Aplicação do *Naïve Bayes* 2 + *Bag-of-words*	
Mensagens identificados correctamente	**Mensagens identificados incorrectamente**
193	**65**

Tabela 24 – Aplicação do *Naïve* Bayes 2 + *Bag-of-words*.

Analisando a Tabela 24 verificamos que esta técnica consegue uma taxa de acerto de aproximadamente de 73%, que é inferior ao valor obtido pelo *Naïve Bayes* 2 e pelo *Bag-of-words* quando testados separadamente. Apesar deste facto foi decidido continuar com o estudo desta técnica para averiguar se realmente iríamos ter resultados mais baixos após a aplicação dos requisitos mínimos. Assim para eliminarmos as mensagens identificadas incorrectamente foi necessário criar requisitos que impedisse que essas mensagens fossem respondidas incorrectamente. Depois de efectuados testes à amostra chegámos à conclusão que as mensagens teriam que respeitar os seguintes requisitos:

<table>
<tr><th colspan="2">Requisitos mínimos</th></tr>
<tr><th colspan="2">Naïve Bayes 2 + Bag-of-words</th></tr>
<tr><td>Resposta simples
Ou
Resposta múltipla
(tipo com maior valor)</td><td rowspan="3">O valor da questão terá que ser maior ou igual a 1.02</td></tr>
<tr><td>Resposta múltipla
(segundo tipo com maior valor)</td></tr>
<tr><td>Resposta múltipla (terceiro tipo com maior valor)</td></tr>
</table>

Tabela 25 – Requisitos mínimos para o *Naïve* Bayes 2 + *Bag-of-words*.

O valor 1.02 apresentado na tabela anterior resulta de testes efectuados sobre a amostra, em que mensagens que apresentassem valores iguais ou superiores a 1.02 nunca seriam identificadas incorrectamente. Obviamente que existem mensagens com valores inferiores que foram identificadas correctamente mas devido a este requisito a sua resposta deixou de ser tratada pelo SiRAC.

Aplicando novamente a técnica de acordo com os requisitos chegamos à Tabela 26 onde verificamos que apenas 91 mensagens são identificadas correctamente, pressupondo uma taxa de acerto de 35%.

Aplicação do *Naïve Bayes* 2 + *Bag-of-words*			
Mensagens identificados correctamente	**Mensagens sem requisitos mínimos**	**Mensagens identificados incorrectamente**	**Total**
91	**167**	**0**	**258**

Tabela 26 – Aplicação do *Naïve* Bayes 2 + *Bag-of-words* com requisitos.

Com estes resultados podemos concluir que a conjugação do método *Naïve Bayes* 2 e o *Bag-of-words*, da forma como foram testados, apenas conseguiu resultados superiores ao *Naïve Bayes* 2 (0%), pois o *Bag-of-words* consegue uma taxa de acerto de 40%.

Não satisfeitos com estes resultados tentamos explorar as configurações do *Bag-of-words* alterando o peso dado aos *tokens* (palavras) e aos conjuntos de *tokens* (expressões). Desta forma foram feitos testes utilizando o *Naïve Bayes* 2 e o *Bag-of-words*, mas agora com diferentes pesos.

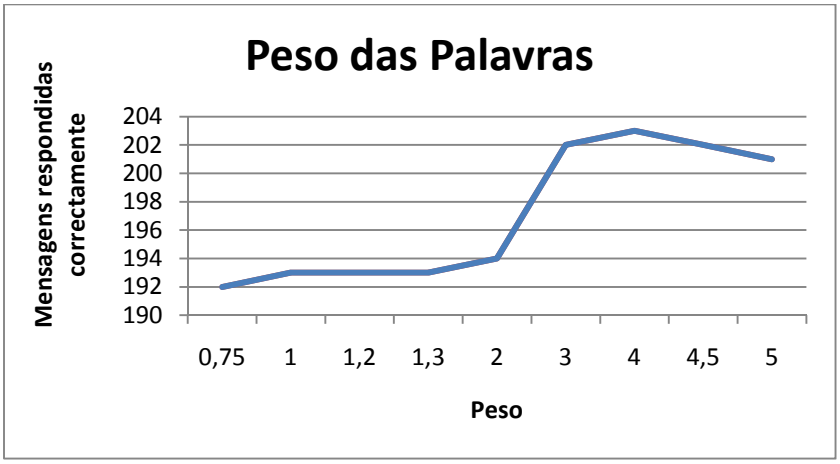

Gráfico 1 – Peso das Palavras do *Naïve Bayes* 2 + *Bag-of-words*.

Analisando o Gráfico 1 podemos constatar que o peso atribuído às palavras para a qual se conseguiram melhores resultados foi 4, verificando-se que correspondia ao máximo obtido no intervalo testado e que a curva tinha um único máximo. Partimos do pressuposto que o valor 4 seria o mais indicado a atribuir ao peso de cada palavra encontrada na mensagem. De seguida foram feitos testes para se calcular o peso mais indicado para as expressões (assumindo o valor de 4 para o peso das palavras).

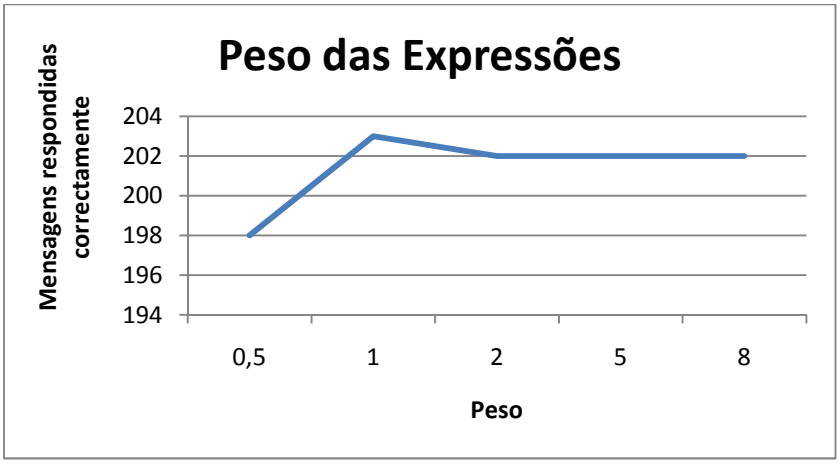

Gráfico 2 – Peso das Expressões do *Naïve Bayes* 2 + *Bag-of-words*.

Analisando o Gráfico 2 constata-se que o peso 1 consegue manter os resultados obtidos com a alteração do peso das palavras, ou seja, não será alterado relativamente ao valor inicial que já era 1. Os outros valores testados reduzem as mensagens identificadas correctamente.

Neste momento estamos perante a técnica *Naïve Bayes* 2 conjugado com *Bag-of-words* mas com pesos diferentes dos originais. Assim foi decidido designar esta nova técnica de *Bag-of-words* de *New*

Technique (*NT* 1), em que o peso das palavras é de 4 e das expressões de 1. Aplicando esta nova conjugação conseguimos os seguintes resultados:

Aplicação do *Naïve Bayes* 2 + *NT* 1	
Mensagens identificados correctamente	**Mensagens identificados incorrectamente**
203	**55**

Tabela 27 – Aplicação do *Naïve* Bayes 2 + *NT* 1.

Analisando para a Tabela 27 constatamos que ainda existem mensagens identificadas incorrectamente e que a taxa de acerto de aproximadamente 79% de nada vale se não eliminarmos as mensagens identificadas incorrectamente. Para isso foi necessário proceder a criação de requisitos apresentados na tabela seguinte:

***Naïve Bayes* 2 + *NT* 1**	
Requisitos mínimos	
Resposta simples Ou Resposta múltipla (tipo com maior valor)	O valor da questão terá que ser maior ou igual a 165
Resposta múltipla (segundo tipo com maior valor)	
Resposta múltipla (terceiro tipo com maior valor)	

Tabela 28 – Requisitos mínimos para o *Naïve* Bayes 2 + *NT* 1.

Para eliminarmos mensagens identificadas incorrectamente foi necessário definir o valor do requisito mínimo em 165, um valor à partida bastante alto e que poderá comprometer os resultados da técnica.

Utilizando o valor definido na tabela anterior como requisito mínimo e aplicando novamente a conjugação das duas técnicas obtemos o resultado seguinte tabela:

Aplicação do *Naïve Bayes* 2 + *NT* 1			
Mensagens identificados correctamente	**Mensagens sem requisitos mínimos**	**Mensagens identificados incorrectamente**	**Total**
1	**257**	**0**	**258**

Tabela 29 – Aplicação do *Naïve* Bayes 2 + *NT* 1 com requisitos mínimos.

Da análise da Tabela 29 constatamos que esta técnica irá responder apenas a 1 mensagem de forma correcta, indo de encontro à "desconfiança" relativamente a um valor tão elevado para o

requisito mínimo (165) e conseguindo resultados piores do que os obtidos com o *Bag-of-words* aplicado isoladamente (taxa de acerto de 40%).

Para comprovar os resultados de seguida iremos aplicar o *Naïve Bayes* 2 + *NT* 1 aos exemplos E1, E2 e E3. Para o E1 destaca-se o tipo correcto (Tipo 5 com 1.17708), para o E2 temos uma resposta errada (deveria ser do Tipo 7), e para o E3 temos um valor superior para o Tipo 2 (0.67290) e o Tipo 5 (1.17708), que são os tipos que identificam correctamente a resposta:

Resultados do *Naïve Bayes* 2 + *NT* 1		
E1	**E2**	**E3**
Tipo 1 => 0.01427 Tipo 2 => 0.03290 **Tipo 5 => 1.17708** Tipo 6 => 0.01469 Tipo 7 => 0.01350 Tipo 8 => 0.04971 Tipo 10 => 0.03569 Tipo 11 => 0.01469	Tipo 1 => 0.10856 **Tipo 2 => 0.76071** Tipo 5 => 0.36109 Tipo 6 => 0.11169 Tipo 7 => 0.21539 Tipo 8 => 0.37807 Tipo 10 => 0.27146 Tipo 11 => 0.11169	Tipo 1 => 0.01427 **Tipo 2 => 0.67290** **Tipo 5 => 1.17708** Tipo 6 => 0.01469 Tipo 7 => 0.01350 Tipo 8 => 0.04971 Tipo 10 => 0.03569 Tipo 11 => 0.01469

Tabela 30 – Aplicação da técnica *Naïve Bayes* 2 + *NT* 1.

Tendo em consideração o valor do requisito mínimo (165) podemos facilmente concluir que esta técnica não conseguirá responder a nenhuma destas mensagens, pois todos os resultados estão bastante abaixo. Estes resultados apenas atestam os dados constantes da Tabela 29.

3.3.6. *Naïve Bayes* 1 + *NT* 2 (*Bag-of-words* com pesos diferentes)

A técnica *Naïve Bayes* 1 + *NT* 2 surge com o objectivo de melhorar os resultados apresentados até ao momento e utiliza um raciocínio semelhante ao da técnica anterior, com a diferença de que o *Naïve Bayes* 1 identifica apenas palavras (ao invés, *Naïve Bayes* 2 identifica palavras e expressões). Comparativamente à técnica anterior será dado maior relevo as palavras relativamente às expressões.

De seguida será aplicado sobre a amostra, a técnica *Naïve Bayes* 1 em conjugação com a *Bag-of-words* em que o resultado será o soma de ambas. A Tabela 31 mostra os resultados:

Aplicação do *Naïve Bayes* 1 + *Bag-of-words*	
Mensagens identificados correctamente	**Mensagens identificados incorrectamente**
189	**69**

Tabela 31 – Aplicação do *Naïve* Bayes 1 + *Bag-of-words*.

A Tabela 31 demonstra que esta técnica consegue identificar correctamente 189 mensagens, ou seja, aproximadamente 73%. Esta taxa de acerto é superior ao *Naïve Bayes* 1 (69%) quando aplicado

isoladamente e inferior ao *Bag-of-words* quando aplicado também isoladamente. Tal como efectuado na técnica anterior, iremos continuar com o estudo de forma a averiguar se será possível aumentar a taxa de acerto. Para isso teremos que criar um requisito mínimo que impeça esta técnica de identificar incorrectamente mensagens.

Depois de efectuados testes à amostra chegámos à conclusão que as mensagens teriam que respeitar os seguintes requisitos:

Naïve Bayes* 1 + *Bag-of-words	
Requisitos mínimos	
Resposta simples Ou Resposta múltipla (tipo com maior valor)	O valor da questão terá que ser maior ou igual a 0.91
Resposta múltipla (segundo tipo com maior valor)	
Resposta múltipla (terceiro tipo com maior valor)	

Tabela 32 – Requisitos mínimos para o *Naïve* Bayes 1 + *Bag-of-words*.

Para que não haja mensagens identificadas incorrectamente o valor mínimo terá que ser 0.91. Todas as mensagens com um valor inferior não serão respondidas pelo SiRAC, mas pelos recursos humanos. De salientar que a utilização de um requisito mínimo implica que mensagens correctamente identificadas mas sem valor suficiente não sejam tratadas pelo sistema.

A Tabela 33 resulta da aplicação da técnica com os requisitos mínimos e onde temos 9 mensagens correctamente identificadas, ou seja, uma taxa de acerto de 3,5%.

Aplicação do *Naïve Bayes* 1 + *Bag-of-words*			
Mensagens identificados correctamente	**Mensagens sem requisitos mínimos**	**Mensagens identificados incorrectamente**	**Total**
9	**249**	**0**	**258**

Tabela 33 – Aplicação do *Naïve* Bayes 1 + *Bag-of-words* com requisitos.

Podemos concluir que a conjugação do método *Naïve Bayes* 1 e o *Bag-of-words* da forma como foram testados não se revelaram uma mais-valia para a resolução do problema proposto e que analogamente ao desenvolvimento da técnica anterior (*Naïve Bayes* 2 + *NT* 1) iremos prosseguir explorando as configurações do *Bag-of-words*. Para isso iremos alterar o peso dado aos *tokens* (palavras) e aos conjuntos de *tokens* (expressões) com o objectivo de maximizar o número de mensagens respondidas correctamente. Desta forma foram feitos testes utilizando o *Naïve Bayes* 1 e o *Bag-of-words*, mas agora com diferentes pesos.

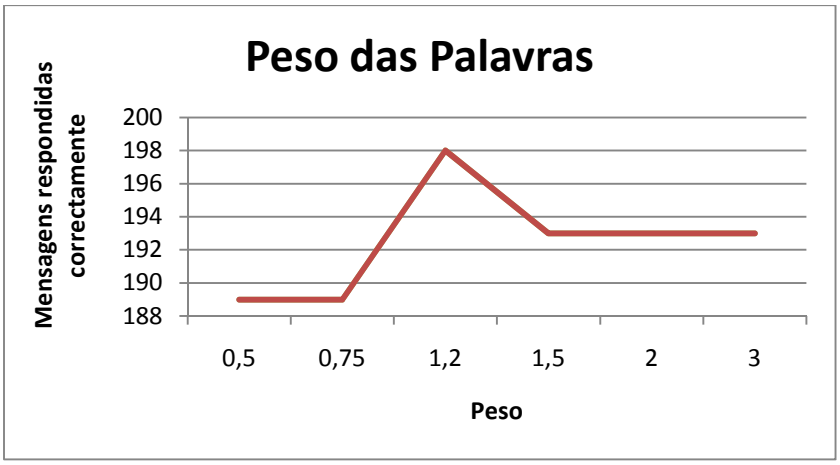

Gráfico 3 – Peso das Palavras do *Naïve Bayes* 1 + *Bag-of-words*.

Analisando o Gráfico 3 concluiu-se que com o peso (das palavras) igual a 1.2 consegue-se o melhor resultado de mensagens identificadas correctamente. Assim, todos os testes feitos a partir deste momento terão em consideração este valor. De seguida será apresentado um gráfico que resulta dos testes efectuadas para o cálculo do melhor peso a atribuir às expressões:

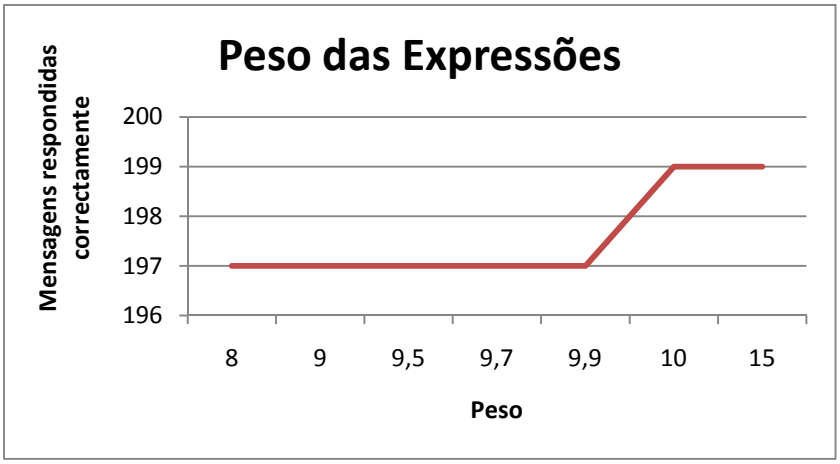

Gráfico 4 – Peso das Expressões do *Naïve Bayes* 1 + *Bag-of-words*.

O Gráfico 4 mostra-nos que o número de mensagens identificadas correctamente consegue o seu maior valor com a partir do peso 10, mantendo-se constante até 15. Optou-se por definir o peso de 10 como valor a utilizar na aplicação da técnica, uma vez que maximiza o número de mensagens com o menor valor possível para o peso.

Neste momento estamos perante a técnica *Naïve Bayes* 1 conjugado com *Bag-of-words* mas com pesos diferentes dos originais. Assim foi decidido designar esta nova técnica de *Bag-of-words* de *New Technique* 2 (*NT* 2) em que o peso das palavras é de 1.2 e das expressões de 10 (conforme comprovado nos gráficos). Aplicando esta nova conjugação conseguimos os seguintes resultados:

Aplicação do *Naïve Bayes* 1 + *NT* 2	
Mensagens identificados correctamente	**Mensagens identificados incorrectamente**
199	**59**

Tabela 34 – Aplicação do *Naïve* Bayes 1 + *NT* 2.

A Tabela 34 demonstra que ainda existem mensagens identificadas incorrectamente e que a taxa de acerto de aproximadamente 77%, apesar de elevada não tem qualquer relevância enquanto não forem eliminadas as mensagens identificadas incorrectamente. Para isso foi necessário proceder a criação de requisitos apresentados na Tabela 35.

Naïve Bayes 1 + *NT* 2	
Requisitos mínimos	
Resposta simples Ou Resposta múltipla (tipo com maior valor)	O valor da questão terá que ser maior ou igual a 0.59
Resposta múltipla (segundo tipo com maior valor)	
Resposta múltipla (terceiro tipo com maior valor)	

Tabela 35 – Requisitos mínimos para o *Naïve Bayes* 1 + *NT* 2.

De acordo com os testes efectuados à amostra foi necessário definir como valor mínimo 0.59, de forma a eliminar todas as mensagens identificas incorrectamente. De salientar que este valor resulta apenas de testes efectuados sobre a amostra fornecida para o desenvolvimento do projecto e que poderá evoluir depois de implementado e/ou no decorrer da utilização.

Utilizando o valor definido na tabela anterior como requisito mínimo e aplicando novamente a conjugação das duas técnicas obtemos o resultado apresentado na Tabela 36:

Aplicação do *Naïve Bayes* 1 + *NT* 2			
Mensagens identificados correctamente	**Mensagens sem requisitos mínimos**	**Mensagens identificados incorrectamente**	**Total**
155	**103**	**0**	**258**

Tabela 36 – Aplicação do *Naïve* Bayes 1 + *NT* 2 com requisitos mínimos.

Analisando a tabela constatamos que esta técnica identifica correctamente 60% das mensagens, conseguindo desta forma o melhor resultado de todas as técnicas utilizadas nesta dissertação. Neste caso a conjugação de 2 técnicas (*Naïve Bayes* 1 e *NT* 2) resultou na melhoria dos resultados devolvidos.

Para comprovar os bons resultados apresentados por esta técnica, de seguida iremos aplicar o *Naïve Bayes* 1 + *NT* 2 aos exemplos E1, E2 e E3. Para o E1 destaca-se o tipo correcto (Tipo 5 com 0.76531), para o E2 temos uma resposta errada (deveria ser do Tipo 7), e para o E3 temos um valor superior para o Tipo 2 (0.59005) e o Tipo 5 (0.86531), que são os tipos que identificam correctamente a resposta:

Resultados do *Naïve Bayes* 1 + *NT* 2		
E1	**E2**	**E3**
Tipo 1 => 0.25029	Tipo 1 => 0.24762	Tipo 1 => 0.25029
Tipo 2 => 0.26725	**Tipo 2 => 0.53881**	**Tipo 2 => 0.59005**
Tipo 5 => 0.76531	Tipo 5 => 0.14873	**Tipo 5 => 0.86531**
Tipo 6 => 0.26725	Tipo 6 => 0.26440	Tipo 6 => 0.26725
Tipo 7 => 0.22087	Tipo 7 => 0.44703	Tipo 7 => 0.22087
Tipo 8 => 0.20807	Tipo 8 => 0.20585	Tipo 8 => 0.20807
Tipo 10 => 0.23489	Tipo 10 => 0.23239	Tipo 10 => 0.23489
Tipo 11 => 0.26725	Tipo 11 => 0.26440	Tipo 11 => 0.26725

Tabela 37 – Aplicação da técnica *Naïve Bayes* 1 + *NT* 2.

Mas agora tendo em conta os resultados da tabela anterior e os requisitos necessários definidos podemos concluir que apenas o exemplo E2 não preenche os requisitos (0.59), ao contrário do E1 e E3 que conseguem identificar as questões existentes nas mensagens (destacadas com sombreado a cinzento).

3.3.7. Conclusões

Após aplicar as diferentes técnicas pudemos verificar que nem todas identificam o tipo de questão correcta e nem todas identificam todas as questões constantes da mensagem de correio electrónico. Um sistema deste género nunca deve enviar uma resposta errada a uma pergunta para não induzir em erro quem tenha enviado a mensagem, pelo que se pretende que o número de mensagens com respostas erradas seja zero. Não se considera tão grave o facto de o sistema não responder à totalidade das questões constantes na mensagem, uma vez que é preferível não fornecer uma resposta do que fornecer uma resposta errada.

Para resolver estas situações foram criados requisitos mínimos para verificar se a mensagem contém as características necessárias para ser considerada como uma resposta relevante de acordo com o respectivo contexto. Ao longo do projecto estes requisitos foram evoluindo de forma a acompanhar o desenvolvimento do SiRAC, em que cada alteração pressuponha novos testes a toda a amostra de mensagens de forma a eliminar mensagens identificadas de forma incorrecta.

Assim para as diferentes técnicas foram necessárias diferentes alterações para se adaptarem à sua forma de classificação. No "*Naïve Bayes* 1"e "*Naïve Bayes* 2" foi necessário distinguir se se tratava de

uma resposta simples ou múltipla. Estas duas técnicas conseguem responder até 3 questões existentes numa mensagem (de acordo com a amostra nunca existem mais do que 3 questões por mensagem). Relativamente à técnica "*Bag-of-words*", à técnica "*Naïve Bayes* 1 + *NT* 2" e à técnica "*Naïve Bayes* 2 + *NT* 1" não foi necessária fazer qualquer distinção relativamente ao número de questões constantes da mensagem.

Quando o mapeamento está de acordo com os requisitos mínimos, ou seja, é maior ou igual ao valor definido, o sistema passa à construção de resposta, caso contrário, reencaminha a mensagem para os recursos humanos para que lhe seja dado o melhor seguimento.

3.4. Construção e envio da resposta

Esta é a última fase antes do envio da resposta para o remetente ou do reencaminhamento da mensagem para ser tratada pelos recursos humanos da Divisão Académica. Este módulo recebe os tipos de questões mapeados na mensagem (através de uma das técnicas anteriores) caso respeitem os requisitos mínimos definidos, procedendo à construção da resposta. Caso não tenha sido identificado nenhum tipo, a mensagem é reencaminhado para os recursos humanos, sendo assinalado no assunto que foi alvo da análise pelo sistema SiRAC.

De forma a enviar, na construção da resposta, uma mensagem o mais personalizado possível foi desenvolvido um módulo que procura nos últimos 10 *tokens* da mensagem recebida pela existência do nome (primeiro e/ou último nome) do remetente. Esta característica tenta criar uma relação de proximidade com o remetente, dando mais confiança em futuras utilizações.

Relativamente ao conteúdo, a mensagem será construída com os *templates* pré-definidos relativos aos tipos de questões mapeados. Os *templates* serão constituídos por toda a informação relevante sobre o tipo de questão a que está associado e de acordo com a época do ano lectivo, pois existem questões que poderão ter respostas diferentes em alturas diferentes do ano. As diferentes respostas a serem enviadas terão que ser actualizadas no sistema pela Divisão Académica.

Na parte final da mensagem é colocado a informação de que se trata de um sistema de resposta automático a mensagens de correio electrónico de forma a informar os utilizadores do serviço que não foram os recursos humanos a responder, mas sim uma aplicação de *software* e que qualquer anomalia deve ser comunicada aos serviços competentes para posteriores correcções e melhorias.

Para comprovarmos o comportamento deste módulo foram utilizadas as mensagens anteriores (E1, E2 e E3), tendo em consideração os requisitos mínimos e com as seguintes respostas:

	Resposta enviada		
	E1	**E2**	**E3**
***Naïve Bayes* 1**	Esta mensagem será reencaminhada para os recursos humanos.		
***Naïve Bayes* 2**	Esta mensagem será reencaminhada para os recursos humanos.		
Bag-of-words	Boa tarde ##### ##### **{*Template* de resposta ao Tipo 5}** Divisão Académica Isto é um sistema de resposta automática a mensagens de correio electrónico.	Esta mensagem será reencaminhada para os recursos humanos.	Boa tarde ##### ##### **{*Template* de resposta ao Tipo 5}** Divisão Académica Isto é um sistema de resposta automática a mensagens de correio electrónico.
***Naïve Bayes* 2 + *NT* 1**	Esta mensagem será reencaminhada para os recursos humanos.		
***Naïve Bayes* 1 + *NT* 2**	Boa tarde ##### ##### **{*Template* de resposta ao Tipo 5}** Divisão Académica Isto é um sistema de resposta automática a mensagens de correio electrónico.	Esta mensagem será reencaminhada para os recursos humanos.	Boa tarde ##### ##### **{*Template* de resposta ao Tipo 2}** **{*Template* de resposta ao Tipo 5}** Divisão Académica Isto é um sistema de resposta automática a mensagens de correio electrónico.

Tabela 38 – Resposta enviada conforme a técnica.

Através da Tabela 38 podemos constatar que o *Naïve Bayes* 1, o *Naïve Bayes* 2 e o *Naïve Bayes* 2 + *NT* 1 não conseguiram construir a resposta a enviar ao remetente, para as diferentes mensagens (E1, E2 e E3), pois não preenchiam os requisitos mínimos definidos. Desta forma seria a Divisão

Académica a proceder ao envio da responda (“`Esta mensagem será reencaminhada para os recursos humanos.`”).

A técnica *Bag-of-words* conseguiu responder correctamente ao exemplo E1, respeitando os requisitos; relativamente ao E2 terão que ser os recursos humanos a responder ao remetente; no E3 apenas é identificado parte da resposta, pois a técnica não conseguiu identificar o Tipo 2. Nos casos em que o sistema apenas consegue responder a parte das questões colocadas, o remetente terá que enviar uma nova mensagem para a Divisão Académica para que seja satisfeita a sua dúvida.

A técnica *Naïve Bayes* 1 + *NT* 2 conseguiu responder correctamente respeitando os requisitos mínimos à mensagem E1 e E3. De salientar que a resposta à mensagem E3 é composta por 2 *templates*, respondendo a 2 questões encontradas na mensagem recepcionado. O E2 terá que ser respondido pelos recursos humanos da Divisão Académica.

De salientar que todas as mensagens enviadas como resposta para os remetentes são também reencaminhadas para a Divisão Académica de forma a haver um registo da resposta dada a um determinada questão. Estes registos poderão servir para averiguar a causa de possíveis incoerências na resposta de mensagens de correio electrónico, ajudando na determinação do problema e a sua resolução.

3.5. Resumo dos Resultados

Depois de aplicar os algoritmos de classificação à amostra de 258 mensagens disponíveis (com 278 questões) e respeitando os respectivos requisitos mínimos (garantem que não se envie uma resposta errada ao remetente) podemos sintetizar os resultados, que se apresentam na Tabela 39.

Resultados da identificação dos Tipos			
Técnica	**Certos**	**Sem Requisitos Mínimos**	**Total**
***Naïve Bayes* 1**	2	256	258
***Naïve Bayes* 2**	0	258	
Bag-of-words	104	154	
***Naïve Bayes* 2 + *NT* 1**	1	257	
***Naïve Bayes* 1 + *NT* 2**	155	103	

Tabela 39 – Resultados da identificação dos Tipos.

Os valores apresentados para o número de respostas certas é afectado devido a um dos objectivos do sistema que consistia em não enviar uma resposta errada ao remetente, pois foi necessário filtrar as

mensagens de forma a eliminar esse tipo de problema. Isto resultou em taxas de acerto mais baixas do que existiam antes de serem filtradas.

Analisando a tabela verificamos que a técnica *Naïve Bayes* 1, o *Naïve Bayes* 2 e o *Naïve Bayes* 2 + *NT* 1 têm taxas de acerto de zero ou muito próximo de zero, o que nos leva a descartá-las para a resolução do nosso problema. Os baixos resultados explicam-se em grande parte pela introdução de requisitos mínimos.

A técnica *Bag-of-words* tem uma taxa de acerto de aproximadamente 40% que acabou por ser uma agradável surpresa, pois é baseado num conceito muito simples em que são contabilizadas as palavras existentes na matriz pré-definida (Tabela 5) e em que a questão inerente à mensagem é mapeada de acordo com o Tipo com mais palavras e expressões encontradas na mensagem.

Tentando aproveitar a simplicidade da técnica *Bag-of-words* e os bons resultados conseguidos isoladamente com as diferentes técnicas, foi decidido conjugar o *Bag-of-words* com o *Naïve Bayes* 1 e com o *Naïve Bayes* 2. De salientar que a técnica *Bag-of-words* sofreu algumas alterações, sendo atribuídos pesos diferentes às palavras e expressões com o objectivo de maximizar o número de mensagens identificadas correctamente, advindo daí o nome *New Technique* (*NT*) em substituição de *Bag-of-words*. Os resultados da conjugação do *Naïve Bayes* 2 com o *NT* 1 trouxeram resultados bastante fracos com uma taxa de acerto de praticamente zero. Esta solução foi desde logo descartada para a resolução do problema.

A conjugação da técnica *Naïve Bayes* 1 com o *NT* 2 conseguiu aproveitar as potencialidades de ambas as técnicas apresentando uma taxa de acerto de aproximadamente 60% relativamente às mensagens identificadas e respondidas correctamente. Esta percentagem acaba por ser um valor bastante bom neste tipo de problema.

Os bons resultados apresentados por esta última técnica justificam a implementação do SiRAC com a técnica *Naïve Bayes* 1 com o *NT* 2 como ferramenta de auxílio aos recursos humanos da Divisão Académica.

4. Conclusão

A resolução do problema proposto, a resposta de forma automática a mensagens de correio electrónico, baseado no contexto, é uma questão com a qual empresas e instituições, que recebem centenas de mensagens semanalmente, se debatem de forma a libertar os recursos humanos para outras tarefas.

Um sistema que respondesse a todas as mensagens de forma correcta seria muito complexo e demasiado ambicioso para o âmbito de um mestrado e não foi o pressuposto da qual partimos. Foi definido desde o início que o objectivo seria responder a determinado tipo de mensagens de forma a auxiliar os recursos humanos da Divisão Académica e nunca como seu substituto.

Dessa forma limitamos o número de tópicos a que o sistema iria responder e concentramo-nos apenas na resolução desses. Isto não quer dizer que no futuro não possam vir a ser acrescentados outros tópicos, pois o contacto com os alunos para este tipo questões exige que se possa evoluir. Essa evolução resulta da dinâmica e diversidade associada às questões do ensino superior.

O SiRAC é um sistema que, enquadrado neste tipo de necessidade, responde a mensagens com determinados tipos de questões que são recorrentemente colocadas à Divisão Académica. É de salientar que o sistema não responde à totalidade das mensagens dos tópicos seleccionados mas

consegue resultados bastante bons relativamente aos tópicos para qual foi desenvolvido. Quando não é possível a construção de uma resposta que satisfaça a mensagem enviada, o sistema reencaminha-a para os recursos humanos para que lhe seja dada o melhor tratamento no envio da resposta.

Sendo um sistema deste género sujeito a alterações relativamente aos tópicos abordados e as respectivas respostas, o SiRAC será um sistema em contínuo desenvolvimento podendo tornar-se uma peça fundamental na interacção do ISEP com o exterior.

5. Trabalho futuro

Como referido anteriormente, o SiRAC é sistema em constante evolução porque exige a introdução manual dos novos tipo de questões e as palavras e expressões que o identificam. Estamos perante uma actividade que poderia ser substituída pelo próprio sistema caso conseguisse identificar novos tipos de questões sugerindo-as ao gestor do sistema que as validaria. Se da validação resultasse um novo tipo de questão o gestor inseriria a devida resposta. Desta forma, a introdução de novas questões tornava-se semi-automático.

Para melhorar a qualidade dos dados enviados na resposta ao remetente e aumentar o leque de questões que o sistema poderia responder seria interessante desenvolver um módulo que possibilitasse interacção com o portal do ISEP. Desta interacção poderia resultar o acesso a informação existente na base de dados relativa ao aluno, como por exemplo: o valor das propinas em debito, resumo das disciplinas em que está inscrito, resumo das faltas, etc., tornando ainda mais personalizadas as mensagens. Refira-se que muitas das questões colocadas não são genéricas, mas sim fortemente dependentes da situação em concreto do aluno que a coloca.

Uma outra possibilidade, também aberta pela interacção com o portal, é a de o próprio sistema actualizar informação no portal, face às solicitações apresentadas na mensagem. Naturalmente que se

entra aqui num campo de complexidade muito mais elevada por todas as questões de segurança que poderia envolver.

6. Referências Bibliográficas

Almeida Hélio Marcos e Ferreira Kecia Aline Correção Automática de Palavras em Textos [Online]. - Junho de 2004. - 10 de Fevereiro de 2009. - http://homepages.dcc.ufmg.br/~nivio/cursos/pa04/seminarios/seminario11/seminario11.html.

Araribóia G. Inteligência Artificial - um curso prático [Artigo]. - Rio de Janeiro, Brasil : [s.n.], 1988.

Bahl L. R., Jelinek F. e Mercer R. L. A maximum likelihood approach to continuous speech recognition. [Jornal]. - [s.l.] : IEEE Trans. Patt. Anal. Machine Intell., 1983.

Bahll L. R. [et al.] A tree-based statistical language model for natural language speech recogrntion. [Jornal]. - [s.l.] : IEEE Trans. AcozMt. Speech Slg. Process., 1989.

Bayes T. An Essay Towards Solving a Problem in the Doctrine of Chances [Relatório]. - [s.l.] : Philos. Trans. R. Soc. London, 1763.

Bodine Jacqueline Improving Bayesian Spelling Correction [Artigo]. - [s.l.] : http://www.cs.cornell.edu/courses/cs674/2005sp/projects/jacqueline bodine.doc, 2005.

Borgelt C. e Gebhardt J. A naive Bayes style possibilistic classifier [Jornal]. - Aachen, Germany : Verlag Mainz, 1999. - 7th European Congress on Intelligent Techniques and Soft Computing.

Borgelt C. e Kruse R. Graphical Models - Methods for Data Analysis and Mining [Relatório]. - United Kingdom : J. Wiley & Sons, Chichester, 2002.

Brown P. F. [et al.] Class-Based n-Gram Models of Natural Language. [Jornal]. - 1990a.

Burger J. e Cardie C. Issues, Tasks and Program Structures to Roadmap Research in Question & Answering [Artigo]. - 2002.

Chakrabarti S. Mining the Web: Discovering Knowledge from Hyper-text Data [Artigo] // Morgan Kaufmann. - 2002.

Cho Sangyeun [et al.] CA-RAM: A High-Performance Memory Subtrate for Search-Intensive Applications [Artigo]. - University of Pittsburgh : [s.n.], 2007.

Clark P., Thompson, J., Porter., B. A Knowledge-Based Approach to Question-Answering [Artigo]. - [s.l.] : In the AAAI Fall Symposium on Question-Answering Systems, 1999.

Domingos P. e Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss [Artigo] // Machine Learning. - 1997.

Friedman N. e Goldszmidt M. Building classifiers using bayesian networks [Relatório]. - [s.l.] : In AAAI/IAAI, Vol. 2, 1996.

Hirschman L., Gaizauskas, R. Natural Language questionanswering: the view from here. [Artigo]. - 2001.

Hirschman L., Light, M., Breck, E. and Burger, J. Deep Read: A reading comprehension system [Artigo]. - [s.l.] : Proceedings 37th Annual Meeting of the Association for Computational Linguistics, 1999.

Jackson Peter Natural Processing for online applications [Jornal]. - Philadelphia : J.Benjamins Pulishing Co., 2002.

Kukich K. Techniques for Automatically Correcting Words in Text. [Artigo]. - [s.l.] : ACM Computing Surveys, 1992. - Vol.24, Nº 4.

Langley P. e Sage S. Induction of selective bayesian classifiers [Relatório]. - [s.l.] : pp. 399–406, 1994.

Lopez Vanessa [et al.] State of the art on Semantic Question Answering [Artigo]. - [s.l.] : Technical Report kmi-07-03, 2007.

Magnini B., Negri, M., Prevete, R., Tanev, H. A WordNet-Based Approach to Named Entities Recognition [Artigo]. - Taipei, Taiwan : Proceedings of SemaNet02, COLING Workshop on Building and Using Semantic Networks, 2002.

Magnini B., Negri, M., Prevete, R., Tanev, H. Is It the Right Answer? Exploiting Web Redundancy for Answer Validation [Artigo]. - Philadelphia : Proceedings of the 40th Annual Meeting of the Association for Computacional Linguistics, 2002.

Maia Roberto Bomeny Detecção da Intrusão utilizando Classificação Bayesiana [Relatório]. - Rio de Janeiro, Brasil : [s.n.], 2005.

Mc Guinness D. Question Answering on the Semantic Web [Artigo]. - [s.l.] : IEEE Intelligent Systems, 2004.

McCallum A. e Nigam K. A comparison of event models for Naive bayes text classification [Artigo] // In Proc. of the AAAI-98 Workshop on learning for text categorization, pp. 41-48.. - 1998.

Medeiros José Carlos Dinis Processamento Morfológico e Correcção Ortográfica do Português [Artigo]. - Lisboa : [s.n.], 1995.

Odell M. K., Russell, R. C. [Artigo]. - Washington, D.C. : [s.n.], 1918.

Oguri Pedro Aprendizado de Máquina para o Problema de Sentiment Classification [Relatório]. - Rio de Janeiro, Brasil : [s.n.], 2006.

Oliveira F. A. D. Processamento de Linguagem Natural: princípios básicos e a implementação de um analisador sintático de sentenças da língua portuguesa [Online]. - 2004. - 12 de 03 de 2009. - http://www.inf.ufrgs.br/procpar/disc/cmp135/trabs/992/Parser/parser.html.

Pacheco H. C. F. Uma Ferramenta de Auxílio à Redação, Dissertação de Mestrado [Artigo]. - [s.l.] : Dcc/UFMG, 1996.

Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference [Relatório]. - [s.l.] : Morgan Kaufmann Publishers Inc., 1988.

Rijsbergen C. J. Van Information Retrieval, 2nd Edtion [Artigo]. - University of Glasgow : Dept. of Computer Science, 1979.

Seo H. Kim and J. A reliable indexing meyhod for a pratical QA system [Artigo]. - [s.l.] : In 19th International Conference on Computacional Linguistics, 2002.

Specia L. Modelagem de um Interpretador Lexical para a Linguagem DART [Artigo]. - Cascavel, Brasil : [s.n.], 2000.

Turba T. N. Checking for spelhng and typographlcal errors in computer-based text. [Jornal]. - [s.l.] : SIGPLANSIGOA Newslett, 1981.

Vieira R. e Strube V. L. Lingüística Computacional: princípios e aplicações [Artigo]. - Rio Grande do Sul, Brasil : [s.n.], entre 2002 e 2004.

Vinhaes Rêges Faria Estudo da Utilização de Técnica de Processamento de Linguagem Natural para Otimização de Tradutores Automáticos [Artigo]. - Rio Verde, Brasil : Universidade de Rio Verde, Faculdade de Ciência da computação, 2005.

Von Wangenheim Aldo Uma Interface em Linguagem Natural de Aplicação Genérica: Estudo de Viabilidade e Implementação [Jornal]. - 1993.

Zipf G. K. the Psycho-Biology of Language [Artigo]. - Boston, EUA : [s.n.], 1935.

Ziviani N. Projeto de Algoritmos com Implementações em Pascal e C [Artigo]. - [s.l.] : Pioneira Thomson Learning, 2004. - segunda edição.