Resolução de Ambiguidades na Normalização de Texto em Português Europeu

Manuel Ribeiro, Daniela Braga, Mário Henriques,
Miguel Sales Dias e Heiko Rahmel
Microsoft

Abstract

Although the research community pays little attention to Text Normalization (TN), this is an essential module in Text-to-Speech and Speech Recognition systems, which has a significant development timeline and requires deep linguistic expertise. One of the big issues is ambiguity resolution in TN, which is particularly problematic when handling numerals in a language. In this paper, we present the major TN challenges we face when developing Speech Technology Systems in European Portuguese and some solutions are proposed, in a rule-based approach. The rules were tested and refined. The overall performance of the current system is 98,50%.

Keywords: text normalization, Text-to-Speech, Speech Recognition, European Portuguese
Palavras-chave: normalização de texto, conversão texto-fala, reconhecimento de voz, ambiguidade, Português Europeu

1. Introdução

A normalização de texto ou pré-processamento é um módulo fundamental dos sistemas de síntese de fala ou TTS (Text-to-Speech) que converte toda a espécie de símbolos, abreviaturas, títulos, siglas, acrónimos, datas, horas, números de telefone, números romanos, fórmulas matemáticas, caracteres especiais, moradas, endereços de e--mail, urls, etc. em sequências ortográficas adequadas para uma subsequente transcrição fonética. A esta operação de conversão chama-se normalização de texto ou TN (Text Normalization). Este módulo é partilhado pelos sistemas de reconhecimento de voz (SR-Speech Recognition), pelo que funciona no sentido inverso, designando-se por ITN (Inverse Text Normalization), uma vez que enquanto nos sistemas de síntese o input é texto e output é voz, nos sistemas de reconhecimento o input é a voz e o output é texto.

A identificação de expressões temporais (datas e horas) e expressões numéricas (monetárias e percentuais), tão necessárias ao Processamento da Fala, é tarefa também tratada na área da Extracção de Informação (Mota et al., 2007). Esta tarefa, conhecida por "named entity recognition" ou reconhecimento de entidades mencionadas, centra-se actualmente na identificação e classificação de nomes próprios, mas em sentido lato

***Textos Seleccionados. XXIV Encontro Nacional da Associação Portuguesa de Linguística*, Lisboa, APL, 2009, pp. 411-426**

também pode designar as entidades que cabem dentro das categorias listadas em cima como típicas de TN e ITN, como siglas, acrónimos, datas, horas, etc. Mota et al. (2007) fazem uma caracterização muito esclarecedora do que se entende por entidade mencionada, técnicas para a sua identificação, principais estudos conduzidos para o Português e iniciativas e resultados de avaliação. O Processamento da Fala poderá beneficiar muito deste conhecimento em trabalhos futuros.

O módulo de TN e ITN enquadra-se na componente de front-end to sistema de TTS e SR respectivamente, o que significa a transformação do texto em etiquetas fonéticas no caso do sistema de TTS e a transformação do sinal de voz em forma ortográfica no caso da arquitectura do SR. É comum que este problema seja resolvido por léxicos fonéticos, ou seja, através de uma lista de entradas e da respectiva expansão ortográfica ou imediata transcrição fonética. Este módulo é dependente da língua, o que faz com que seja muito específico e que necessite de permanente actualização, dado que o léxico das línguas está em permanente expansão. Apesar de este assunto ser geralmente tido como trivial pela comunidade académica, o que explica que lhe seja dedicada pouca atenção (Teixeira, 1995; Oliveira, 1996; Teixeira, 2004), a verdade é que se trata um módulo que implica cerca de dois meses de desenvolvimento por um especialista a tempo inteiro e um mês de teste e refinamento, quando o objectivo é a sua integração em sistemas de síntese de fala destinados a serem comercializados. No que se refere à normalização de texto em português, são de salientar os estudos de Braga (2008), Trancoso & Viana (1995), Trancoso & Viana (1997), Barbosa et al. (2003). Não é habitual, por exemplo, encontrar descrições e soluções para os vários casos de ambiguidade que ocorrem ao nível deste módulo e para os quais é necessário propor regras de desambiguação. Um dos principais problemas relaciona-se com a dificuldade em identificar os números de telefone como tal e não como um número cardinal. Outros problemas surgem na leitura das seguintes expressões: <século XIV> e <XIV Festival Internacional de Teatro de Expressão Ibérica>, em que o número romano <XIV> se lê no primeiro caso como numeral cardinal <catorze> e no segundo caso como numeral ordinal <décimo quarto>. O mesmo ocorre na leitura de números romanos, que, sendo representados por letras, criam situações de ambiguidade com siglas. Existem ainda casos em que se verifica homografia em títulos, como é o caso de <D.>, que pode ser expandido como <Dom> ou <Dona>, em função do género do nome que lhe segue. A leitura de dígitos é talvez o problema mais complexo, na medida em que é necessário reconhecer padrões de leitura para horas, datas, unidades monetárias, expressões matemáticas, números de telefone e simples numerais cardinais e ordinais. O trabalho que aqui se apresenta organiza-se da seguinte forma: na secção 2 descreve-se o processo de organização e estrutura das regras; na secção 3 expõem-se os principais problemas ao nível da ambiguidade em TN e ITN e propõem-se soluções; na secção 4 apresentam-se os testes e resultados obtidos e na secção 5 as conclusões e trabalho futuro.

2. Regras de TN

2.1. Estrutura e Construção das Regras de Normalização de Texto

As regras de normalização de texto estruturam-se a partir de uma linguagem derivada do SSML (Speech Synthesis Markup Language)[1], designada por TNML[2] e adaptada ao módulo de TN. O SSML foi recomendado pelo W3C[3] (World Wide Web Consortium) desde 2004 e é uma das linguagens de mark-up mais utilizadas actualmente. Embora o TNML não seja de difícil percepção, a acumulação de regras e a constante referenciação torna o processo demasiado intricado para se trabalhar num editor de texto. Desta forma, a escrita das regras de normalização de texto foi facilitada pela criação de uma ferramenta interna que estabelece uma ligação entre as regras e a linguagem em que são escritas. A estrutura de um mapa de TN assemelha-se à de um esquema em árvore, com sucessivas referências e posições de acordo com a função, a saber: 1) Terminais, 2) Regras Top-Level, 3) Regras Sequenciais e 4) Regras em Lista. A Figura 1 procura exemplificar graficamente de uma forma simplificada a estrutura das regras de TN.

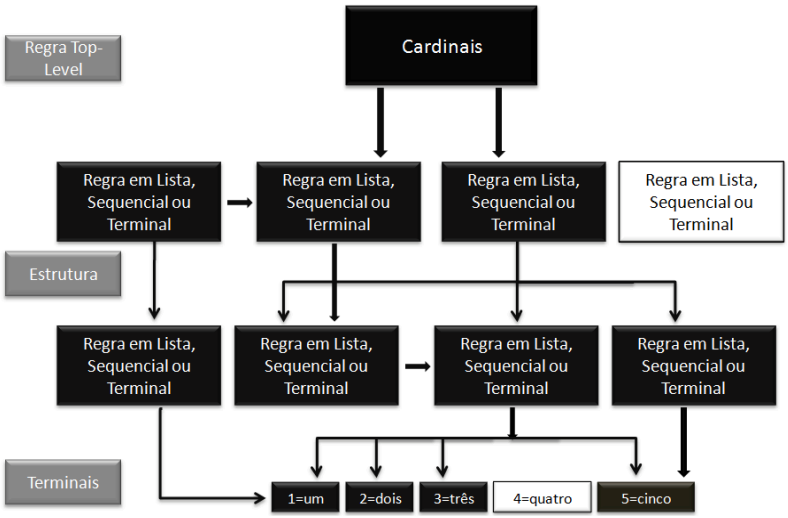

Figura 1: Estrutura das regras de TN

[1] Informação disponível em http://www.w3.org/TR/speech-synthesis/ (26/2/2009).

[2] Esta linguagem de programação, bem como todo o processo de edição e teste das regras apresentado na secção 2, estão protegidos por patente: Patent Serial No. 12/361,114.

[3] Sobre o W3C: http://www.w3.org/ (26/2/2009).

2.1.1. Terminais

Os terminais são o elemento menor de um mapa de normalização de texto e aquele que se encontra mais abaixo na estrutura das regras. São as regras que contêm a informação a normalizar. A informação que <1> se normalizará como <um> ou <uma> estará contida num terminal.

2.1.2. Regras Top-Level

Ao contrário dos terminais, as regras Top-Level são o elemento maior. É para as regras Top-Level que todas as restantes regras confluirão e são estas a porta de entrada para um mapa de TN. Ao dar início ao processo de normalização, o sistema começará sempre por uma regra Top-Level. Assim, todas as regras deverão estar associadas, directa ou indirectamente, a uma regra Top-Level. Na Figura 1, o terminal <4=quatro> não se liga à estrutura principal e não entra na regra <Cardinais>, pelo que numa normalização, nunca seria acedido ou identificado pelo sistema. O mesmo se aplica às restantes regras, quer sejam terminais ou de outro tipo.

2.1.3. Regras em Lista

As regras em lista adquirem este nome por representarem as suas referências verticalmente. A principal característica é a de permitir apenas uma normalização das várias disponíveis. É possível, assim, agrupar os elementos de forma a poderem ser reutilizados ou referenciados por outras regras, em que apenas um dos elementos é seleccionado. Veja-se, para melhor entendimento, a representação da Figura 2.

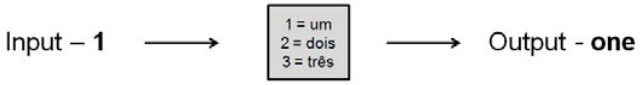

Figura 2: Regra em lista 1 a 3

2.1.4. Regras Sequenciais

As regras sequenciais dispõem os elementos horizontalmente e marcam-se pela obrigatoriedade de todas as suas referências. Trata-se, na verdade, de uma concatenação dos elementos, em que o output é a soma de todos eles, como mostra a Figura 3.

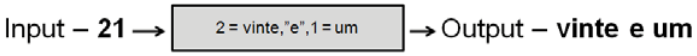

Figura 3: Regra sequencial

2.1.5. Outros Elementos

Além destes quatro tipos de regras, é possível utilizar o TNML para trabalhar com outros elementos, que permitirão um melhor controlo sobre os padrões previstos e as respectivas normalizações, nomeadamente:

– *Espaços*: introdução de Espaços onde não existiam, como em <21> = <vinte e um>, ou a sua remoção.

– *Prioridades*: assumindo um mapa para TN (usado em síntese de fala) e ITN (usado em reconhecimento de fala), as prioridades permitem atribuir um valor para desambiguar expressões idênticas. É o caso de, por exemplo, <1º> = <primeiro> e <1.º> = <Primeiro>.

– *Concordância*: a concordância permite às regras de TN fazer a ponte com um léxico anotado e extrair dele informação que ajudará à desambiguação de casos específicos. Em português, esta é a forma de desambiguar a variação em género (cf. 3.1.1. e 3.2.1.).

– *Reordenação*: A reordenação, em português, não tem um impacto forte na desambiguação, mas é útil para a inteligibilidade. Permite uma alteração da ordem do input e do output. Numa sequência como <2009/01/01>, é possível obter o output <um de Janeiro de dois mil e nove>.

As regras de normalização de texto formam-se estruturando estes elementos das mais variadas formas, de modo a obter o resultado esperado. A correcta utilização dos quatro tipos de regras e os variados elementos permite a construção de um mapa de normalização de texto que produza resultados aceitáveis.

Assim, devido à sua crescente complexidade, a escrita das regras de TN deve iniciar-se pelo elemento menor, os terminais, e afunilar em direcção ao elemento maior, as regras Top-Level. Segue-se a realização de uma série de testes de forma a encontrar erros, que serão corrigidos numa noutra versão do mapa de TN. Trata-se, assim, de um processo iterativo de escrita, teste e afinação das regras, tornado claro na secção 4 e exemplificado na Figura 4.

Figura 4: Processo de construção de regras de normalização de texto

2.2. Domínios da Normalização de Texto em Sistemas de Síntese e Reconhecimento de Fala

O módulo de normalização de texto varia conforme as suas aplicações. Para síntese e reconhecimento de fala, estabeleceram-se vários domínios sobre os quais incidem as regras. Estes domínios associar-se-ão às regras Top-Level, funcionando cada um de forma independente. Embora as regras se auto-referenciem e os domínios se cruzem, o facto de serem Top-Level permite que ganhem um estatuto independente, não sendo influenciadas pelo comportamento das restantes (exceptuando os casos em que existem inputs idênticos (cf. 3.1.7. e 3.2.7.).

As categorias previstas no módulo de normalização de texto são as seguintes: cardinais, percentagem, operações matemáticas, data, tempo, medidas, unidades monetárias, números de telefone, intervalos, números romanos, ordinais, fracções, temperatura, títulos, moradas, domínios Web (emails, URLs, File Paths).

3. Resolução de Ambiguidade

3.1. Principais Problemas

A simples construção de regras usando a tipologia referida na secção 2 não é suficiente para se obterem os resultados esperados, existindo inúmeras situações em que uma relação input/output não chega para uma normalização correcta.

Os casos aqui apresentados foram pensados sempre no sentido do texto para fala (TN), embora possam ser aproveitados também no sentido do reconhecimento de voz, ou seja, da fala para o texto (ITN). Segue-se uma lista dos principais problemas seguidos de um exemplo ilustrativo.

3.1.1. Cardinais: concordância em género

a) 1 = um

Ex. Até agora, as equipas apenas podiam utilizar camisolas numeradas de **1** a 11 e sem indicação dos nomes.

b) 1 galinha = uma galinha

Ex. O Arsenal (que começou por tremer muito) tem 4 vitórias, 1 empate e **1** derrota em casa.

c) 1 galo = um galo[4]

Ex. O Arsenal (que começou por tremer muito) tem 4 vitórias, **1** empate e 1 derrota em casa.

[4] Cf. 3.2.1 acerca da diferença entre os exemplos a) e c).

3.1.2. Números Romanos/Leitura de Letras

a) X = x (letra)

Ex. A «reportagem imaginária» do Repórter **X** saiu nas páginas de três números sucessivos da revista, entre Janeiro e Fevereiro de 1929.

b) X = décimo

Ex. O **X** Encontro Juvenil de Ciência principiou ontem em Lisboa, com uma sessão solene nos Paços do Concelho.

3.1.3. Cardinais/Números de Telefone

a) 961234567 = novecentos e sessenta e um milhões, duzentos e trinta e quatro mil, quinhentos e sessenta e sete.

Ex. O resultado da conta foi de **961234567**.

b) 961234567 = nove seis um dois três quatro cinco seis sete.

Ex. O engenheiro estará contactável no **961234567**.

3.1.4. Datas/Fracções/Cardinais

a) 10/10 = dez de Outubro

Ex. O António faz anos a **10/10**.

b) 10/10 = dez décimos

Ex. No final, apostou apenas 1/10 e acabou por ficar com **10/10**.

c) 10/10 = dez de dez

Ex. Página **10/10**.

3.1.5. Medidas/Tempo

a) 1 m. = um metro

Ex. Descontando os saltos, ela tem cerca de **1 m** e 50 cm de altura…

b) 1m. = um minuto

Ex. Bugno, que ocupa a quinta posição na geral-individual, a **1 m** 57 s de Chiccioli, vê, assim, serem reduzidas as suas hipóteses de vencer o «Giro» pela segunda vez consecutiva.

3.1.6. Cardinais/Operações Matemáticas/Intervalos

a) -1 = menos um

Ex. Subtraindo 1 a 0, ficamos com **-1**.

b) 1-1 = um menos um[5]

[5] Embora os casos a) e b) possuam a mesma normalização, trata-se de domínios diferentes (cardinais e operações matemáticas), que apresentam padrões diferentes.

Ex. Muitos não sabem quanto é **1-1**.
c) 1-1 = um a um
Ex. Ontem conseguiu um empate (**1-1**) em Vila do Conde, frente ao Rio Ave.

3.1.7. Datas/Cardinais[6]

a) 1995 = mil novecentos e noventa e cinco (Data)
Ex. Em Portugal, o número de utilizadores da Internet passou de 5.000 no início de **1995** para 20.000 no início deste ano.
b) 1995 = mil novecentos e noventa e cinco (Cardinal)
Ex. O primeiro número que lhe veio à cabeça foi **1995**.

3.2. Soluções Propostas

As regras a seguir apresentadas foram construídas tendo por base o critério de frequência de ocorrência na língua, devidamente validado em corpora de texto reais, como o Cetem-Público, entre outros obtidos internamente.

3.2.1. Cardinais: concordância em género

O estabelecimento de uma ponte entre as regras de normalização de texto e um léxico anotado com informação de género permite solucionar o problema da concordância em género. Restringindo a forma masculina e feminina do numeral a serem normalizadas apenas quando seguidas por nome ou adjectivo possibilita uma normalização correcta. Uma terceira regra introduz o numeral desprovido de qualquer contexto a ser normalizado apenas quando nenhuma das primeiras se verificar. Assim, <1> = <uma> quando seguido de nome ou adjectivo feminino e <1> = <um> quando seguido de nome ou adjectivo masculino. De outra forma <1> será sempre igual a <um>.

3.2.2. Números Romanos/Leitura de Letras

De forma a desambiguar os números romanos, tomou-se a leitura de letras como padrão e criou-se uma biblioteca de contexto que forçará a normalização dos caracteres como números romanos, quer seja a sua forma cardinal ou ordinal. A Tabela 1 reflecte a contagem desta biblioteca de contexto, separada por domínios. Assim, existem cinquenta e três nomes próprios masculinos que permitem a normalização de números romanos.

[6] Os casos apresentados em 3.1.7. mantêm a mesma relação input/output, mas originam ambiguidades a nível do domínio (cf. 3.2.7).

Domínio	Contagem	Exemplo
Nomes Próprios Masculinos	53	Bento, Felipe, Manuel
Nomes Próprios Femininos	7	Maria, Beatriz, Vitória
Abreviaturas Masculinas	8	séc., art., vol.
Contexto Geral Masculino	45	Congresso, Capítulo, Simpósio
Contexto Geral Feminino	41	Maratona, Corrida, Edição
Outras Formas	3	a.C., d.C., D.
Total	157	

Tabela 1: Biblioteca de desambiguação de números romanos

Exemplos:

a) "Senti igualmente uma grande admiração pela sabedoria de **Luís XIV** e **Frederico II**, quando dinamizaram na Europa e na Prússia a construção de canais que interligaram grandes rios europeus" ("Luís catorze", "Frederico segundo").

b) "Militantes comunistas encontraram-se em público e exigiram a antecipação do **XIV Congresso**" ("décimo quarto Congresso").

3.2.3. Cardinais/Números de Telefone

Estabeleceram-se prioridades sobre os números de telefone, invalidando todos os cardinais de nove dígitos que se iniciem pelos prefixos telefónicos utilizados em Portugal. Embora não se trate de uma fórmula de desambiguação que permita a coexistência de duas formas idênticas, a sequência de testes indicou que a frequência destes cardinais sobre os números de telefone é muito menor. Assim, dando prioridade aos números de telefone sobre os cardinais, sempre que uma sequência de dígitos sem espaços se inicie por um prefixo telefónico comum, produzirá uma normalização dígito por dígito. Exemplo: 961234567 = nove seis um dois três quatro cinco seis sete.

3.2.4. Datas/Fracções

De forma a desambiguar os cardinais das fracções e datas, foi criada uma biblioteca de contexto que normalizará a sequência de caracteres sempre como cardinais (Tabela 2).

Biblioteca de Contexto	Normalizações	Biblioteca de Contexto	Normalizações
bl.	bloco	bl.	bloco
bloco	bloco	págs.	páginas
email	email	p.	página
e-mail	email	pág.	página
nível	nível	pg.	página
mail	mail	página	página
mens.	mensagem	semestre	semestre
msg.	mensagem	ano	ano
mensagem	mensagem		

Tabela 2: Biblioteca de contexto para desambiguação de cardinais separados por barra

Desta forma, sempre que ocorrer o padrão [0-999]/[0-999] precedido por uma das palavras ou abreviaturas, será normalizado para a forma cardinal, sendo introduzida a preposição <de>. Exemplos:

a) página 10/10 = página dez de dez
b) e-mail 2/5 = e-mail dois de cinco
c) msg. 134/459 = mensagem cento e trinta e quatro de quatrocentos e cinquenta e nove

A desambiguação entre datas e fracções torna-se mais difícil, pelo que se optou por dar prioridade na normalização das datas. Assim, num contexto [1-31]/[1-12], a normalização tomará a forma de uma data.
Exemplos:

a) 31/1 = trinta e um do um
b) 05/10 = cinco do dez
c) 10/10 = dez do dez

3.2.5. Medidas/Tempo

A ocorrência de dígitos com a abreviatura <m.> torna possível a normalização em <minuto>, <minutos>, <metro> ou <metros>. Embora a ocorrência de um dígito singular possa diferenciar facilmente a normalização para singular ou plural, a diferenciação entre <minutos> e <metros> torna-se mais complicada. Desta forma, para permitir a desambiguação, eliminou-se a ocorrência de <m.> como abreviatura para minutos quando ocorre isoladamente. Pressupõe-se, nestes casos, a utilização de abreviaturas mais comuns como <min.> ou <mins.>. Quando em contexto de horas e/ou segundos, a abreviatura <m.> é aceitável e normalizada para <minuto(s)>.
Exemplos:

a) 1m. = um metro
b) 1 min. = um minuto
c) 20h15m. = vinte horas e quinze minutos

3.2.6. Cardinais/Operações Matemáticas/Intervalos

De forma a desambiguar cardinais e operações matemáticas de intervalos, utilizaram-se duas palavras que possibilitam uma fácil desambiguação: <de> e <entre>. E, com base nos testes realizados (cf. Secção 4), conclui-se que a frequência das operações matemáticas não é tanta como a dos cardinais (em resultados desportivos, por exemplo). Desta forma, o padrão "[0-200]-[0-200]" será sempre normalizado como cardinal e não como operação matemática. Qualquer contexto adicional, como símbolos matemáticos, possibilitará uma normalização matemática.
Exemplos:

a) de 1-3 = de um a três
b) entre 23-30 = entre vinte e três e trinta
c) 2-2 = dois dois
d) 2-2=0 = dois menos dois é igual a zero

3.2.7. Datas/Cardinais

Aplicação	Biblioteca de Contexto
dia	dia, Dia, dias, Dias
mês	Mês, mês, Mês de, mês de
ano	Ano de, Ano, Férias de, Primavera de, Páscoa de, Natal de, Inverno de, Verão de, Outono de, a partir de, ano, ano de
ano/mês	finais de, final de, até, desde, início de, em, A partir de, Após, Em, Até, Desde, Início de, Final de, Finais de, após
ano/mês/dia	período de, Período de

Tabela 3: Biblioteca de desambiguação de datas e cardinais

Embora o input e o output sejam idênticos, não havendo ambiguidade a nível da normalização, existe uma ambiguidade a nível da categoria (cf. 3.1.7). Desta forma, é possível a criação de uma outra Biblioteca de Contexto que associará ocorrência e normalizações de cardinais a uma data. Além da óbvia desambiguação com a precedência de dias e/ou mês, criaram-se outras para identificar anos, meses e dias em contextos específicos. A Tabela 3 mostra a Biblioteca de Contexto criada, enquanto que a Tabela 4 os contextos previstos para cada aplicação.

Contextos Previstos	
ano	[1500-2100]
	[1500-2100]/[1500-2100]
	[1500]/[00-99]
mês	[mês][ano]
dia	[dia][mês][ano]
	[dia][mês]
	[dia]/[dia][mês][ano]

Tabela 4: Contextos previstos para a desambiguação de datas a partir da biblioteca de desambiguação prevista na Tabela 3

4. Testes e Resultados

Na construção do módulo de normalização de texto para português europeu foram conduzidos, durante o processo, quatro tipos de testes: Testes às Regras de TN na Ferramenta Interna; Testes às Regras de TN sobre um Corpus; Testes ao Módulo de TN e Testes de Inteligibilidade.

4.1. Testes às Regras de TN na Ferramenta Interna

Os primeiros testes são feitos a par da escrita e afinação das regras de normalização. Trata-se de testes simples (input/output) sem qualquer tipo de contexto e normalmente associados a uma regra Top-Level ou a regras menores. A função destes testes preliminares é a de avaliar a funcionalidade das regras à medida que se criam e eliminam pequenos erros que surjam na concatenação e referenciação, como espaços a mais ou em falta, problemas de concordância, incorrecções nas referências ou estruturas, etc. Embora o número possa variar consoante a regra, estipulou-se um número mínimo de nove casos para cada regra Top-Level. Outros casos foram pontualmente acrescentados a sub-regras de forma a testar a sua eficácia durante o processo de escrita. A forma de validação destas regras ocorre por um processo de passed/failed, pelo que o a taxa de acerto destes testes preliminares tem de ser 100% para se poder passar à fase seguinte.

4.2. Testes às Regras de TN sobre um Corpus

O segundo nível de testes pressupõe uma versão beta do mapa de normalização de texto, com todos os domínios finalizados e os testes preliminares completos (i.e., todos os testes "passed"). Através de uma segunda ferramenta, as regras de TN são aplicadas sobre um corpus por normalizar, originando dois ficheiros: o corpus normalizado e um ficheiro estatístico. Após a análise dos resultados, volta-se às regras de forma a acrescentar padrões não previstos e a corrigir erros apontados. Na construção do mapa de normalização de

texto, este processo foi feito por duas vezes, tendo como objectivo a obtenção e a análise de c. 20 000 normalizações.

4.2.1. Primeira Iteração

Para esta primeira iteração, o corpus utilizado continha cerca de 60 000 frases retiradas aleatoriamente do Cetem-Público do qual se obtiveram cerca de 20 000 normalizações, das quais cerca de 10000 foram analisadas e distribuídas pelos possíveis domínios (cf. Tabela 5).

Domínios	**Normalizações**
Cardinais	7779
Percentagem	26
Operações Matemáticas	275
Data	170
Tempo	679
Medidas	188
Unidades Monetárias	7
Números de Telefone	0
Intervalos	30
Números Romanos	0
Ordinais	347
Fracções	43
Temperatura	0
Títulos	43
Moradas	399
Domínios Web	6
Total	**9992**
Taxa de Erro	8,2%

Tabela 5: Distribuição das normalizações na primeira iteração

Cada normalização foi analisada individualmente e em contexto. Os principais erros foram agrupados e corrigidos nas regras de normalização de texto. A taxa de erro de TN representa a percentagem de normalizações consideradas como erradas devido a problemas nas regras, não se incluindo inconsistências do corpus (erros ortográficos, etc.).

4.2.2. Segunda Iteração

Para a segunda iteração, utilizou-se um corpus com cerca de 30 000 frases retiradas aleatoriamente de um corpus interno em português europeu e obtiveram-se cerca de 70 000 normalizações, das quais 10 000 foram analisadas e distribuídas pelos vários domínios (cf. Tabela 6).

Domínios	Normalizações
Cardinais	5793
Percentagem	0
Operações Matemáticas	206
Data	910
Tempo	1681
Medidas	98
Unidades Monetárias	0
Números de Telefone	0
Intervalos	41
Números Romanos	15
Ordinais	1135
Fracções	0
Temperatura	0
Títulos	3
Moradas	104
Domínios Web	4
Total	**9990**
TN error rate	3,5%

Tabela 6: Distribuição das normalizações na segunda iteração

Cada normalização foi analisada individualmente e em contexto. Os principais erros foram agrupados e corrigidos nas regras de normalização de texto. A taxa de erro de TN representa a percentagem de normalizações consideradas como erradas devido a problemas nas regras, não se incluindo inconsistências do corpus (erros ortográficos, etc.).

4.3. Testes ao Módulo de TN

Os testes ao módulo de normalização de texto procuram não apenas a detecção de erros, mas também a análise do seu comportamento dentro de um sistema de síntese de fala, operando juntamente com os restantes módulos.

Desta forma, foram reunidos 124 casos de teste, distribuídos pelos vários domínios (cf. Tabela 7), e submetidos ao sistema por uma plataforma de teste interna. Seguindo o sistema proposto na Figura 4, os erros foram categorizados, analisados e corrigidos, pelo que a taxa de acerto final é de 100%.

Domínios	Número de Casos
Cardinais	30
Percentagem	0
Operações Matemáticas	0
Data	8
Tempo	9
Medidas	8
Unidades Monetárias	13
Números de Telefone	5
Intervalos	0
Números Romanos	9
Ordinais	17
Fracções	6
Temperatura	0
Títulos	4
Moradas	8
Domínios Web	7
Total	**124**

Tabela 7: Testes ao Módulo de TN

4.4. Testes de Inteligibilidade

Os testes de inteligibilidade foram criados com o objectivo de testar certas dimensões mais problemáticas do módulo de TN integrado em todo o sistema de síntese de fala. Foram sintetizadas 150 frases a partir de textos reais, cobrindo quatro domínios específicos de TN (moradas, data, tempo e números de telefone), mas abrangendo outras categorias. Os testes foram conduzidos por sete pessoas (três mulheres e quatro homens).

Domínios	Taxa de inteligibilidade
Moradas	95,7%
Números de Telefone	99,3%
Datas e Tempo	98,2%
Parágrafos	99,6%
Frases	99,7%
Total (média)	**98,50**

Tabela 8: Resultados dos testes de inteligibilidade

5. Conclusões

Neste trabalho, apresentou-se a descrição da estrutura das regras de TN e ITN em português europeu, com o foco nos principais desafios subjacentes à tarefa de resolução de ambiguidades linguísticas. Não sendo um módulo pelo qual a comunidade académica se interesse tradicionalmente, é talvez aquele que apresenta o ciclo de desenvolvimento e estabilização mais longo de todos os módulos de front-end de sistemas de síntese e reconhecimento de voz, precisamente pelo grande impacto que tem quer ao nível da taxa de inteligibilidade na síntese da fala quer ao nível taxa de reconhecimento. Assim, o mapa de regras proposto foi alvo de um longo processo de testes iterativos, culminando num teste de inteligibilidade apenas no sentido da síntese da fala (sentido TN e não ITN) no qual se obteve 98,50% de acerto ao nível da palavra e da sequência normalizada. A obtenção de um mapa de normalização de texto para sistemas de síntese e reconhecimento de fala em qualquer língua com uma taxa de acerto de 100% é muito difícil ou mesmo impossível. Subsistirão sempre padrões que não foram previstos ou que é preciso actualizar, visto que a língua é um organismo vivo em constante evolução. Assim, a construção de um mapa de TN é um trabalho em curso e será sempre um processo iterativo, determinado apenas por cenários, prazos e objectivos. Como trabalho futuro, pretende-se avaliar o impacto do módulo de ITN no reconhecimento de voz em português europeu.

Referências

Barbosa, Filipe Leandro, Maria Carlota Rosa, Carlos Alexandre Gonçalves, Fernando Gil Vianna Resende Jr. (2003) Algoritmo para leitura de siglas em um sintetizador. *Anais do XX Simpósio Brasileiro de Telecomunicações*. Rio de Janeiro: IME/PUC-Rio, pp. 672-675.

Braga, Daniela (2008) Algoritmos de Processamento da Linguagem Natural para Sistemas de Conversão Texto-Fala em Português. Dissertação de Doutoramento, Universidade da Coruña, Espanha.

Mota, Cristina, Diana Santos & Elisabete Ranchhod (2007) Avaliação de Entidades Mencionadas: Princípio de AREM. In Diana Santos (ed.) *Avaliação Conjunta: Um novo paradigma no processamento computacional da língua portuguesa*. IST Press, 2007. ISBN: 978-972-8469-60-8.

Oliveira, Luís Caldas (1996) *Síntese de Fala a Partir de Texto*. Dissertação de Doutoramento, Universidade Técnica de Lisboa.

Teixeira, João Paulo (1995) *Modelização Paramétrica de Sinais Para Aplicação em Sistemas de Conversão Texto-Fala.* Dissertação de Mestrado, Faculdade de Engenharia da Universidade do Porto.

Teixeira, João Paulo (2004) *A Prosody Model to TTS Systems*. PhD Thesis, Faculdade de Engenharia da Universidade do Porto.

Trancoso, Isabel & Maria Céu Viana (1995) Issues in the Pronunciation of Proper Names: the experience of the Onomastica project. *Workshop on Integration of Language and Speech*. Moscow, Russia.

Trancoso, Isabel & Maria Céu Viana (1997) On the pronunciation mode of acronyms in several european languages. *Eurospeech 1997, 5th European Conference on Speech Communication and Technology*. Rhodes, Greece. pp. 573-576.