Capítulo 10

R3M, uma participação minimalista no Segundo HAREM

Cristina Mota

Cristina Mota e Diana Santos, editoras, *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, 2008, Capítulo 10, p. 181–193.

O sistema R3M apresentou-se no Segundo HAREM como um sistema de reconhecimento de pessoas, organizações e locais. Optámos por nos cingir a esta categorias, dado que, de uma forma geral, têm sido mais extensivamente estudadas na área de extracção de informação, e não tínhamos disponibilidade de dedicar mais tempo ao desenvolvimento do nosso sistema.

No entanto, o sistema R3M foi desenhado de modo a que fosse flexível, permitindo no futuro estender facilmente o reconhecimento a outras categorias, assim como incluir o reconhecimento de relações entre entidades mencionadas. Além de ser flexível, o sistema caracteriza-se também por fazer um uso mínimo de recursos linguísticos construídos manualmente, sejam estes regras ou textos anotados. Este último critério resulta do facto de tanto regras como textos anotados criados manualmente serem dispendiosos e morosos de obter, como já argumentado por diversos autores que, normalmente, optam por métodos de aprendizagem semi-supervisionados (Ji e Grishman, 2006; Collins e Singer, 1999; Miller et al., 2004) e não supervisionados (Etzioni et al., 2005).

Assim, o nosso sistema assenta numa estratégia de aprendizagem semi-supervisionada que recorre a um algoritmo de co-treino para inferir regras de classificação (Collins e Singer, 1999). O algoritmo de co-treino que Collins e Singer (1999) apresentam tem a grande vantagem de obter bons resultados de classificação que rondam os 80% de correcção (em inglês, *accuracy*) usando apenas um número muito reduzido de exemplos previamente anotados.

Salientamos desde já que a estratégia proposta por estes autores foi aplicada com sucesso ao problema de REM em textos escritos em português por Mota (2009)[1], que introduziu diversas modificações com vista a obter um anotador de entidades em texto e não apenas um classificador de listas de entidades[2]. O sistema R3M é pois uma reimplementação do sistema criado por Mota (2009), apresentando em relação a este diversas melhorias.

Neste capítulo começamos por descrever o sistema R3M, destacando as melhorias que fomos introduzindo (secção 10.1) relativamente ao sistema em que nos inspirámos. Em seguida, na secção 10.2, mostraremos e analisaremos os resultados da nossa participação. Concluímos o capítulo (secção 10.3) mencionando aspectos positivos e negativos da nossa participação no Segundo HAREM.

10.1 Descrição do sistema R3M

A arquitectura geral do sistema R3M, ilustrada na figura 10.1, é idêntica à do sistema implementado por Mota (2009), a qual foi inspirada, como já referimos, na proposta de Collins e Singer (1999).

Muito sucintamente, como se pode ver na figura, trata-se de um sistema modular sequencial, que separa a fase de identificação de entidades mencionadas da sua classificação. Pode ver-se, igualmente, que o sistema envolve uma fase de treino, em que aprende regras de classificação com base num algoritmo de co-treino, e uma fase de teste que usa as regras

[1] Tal como discutido pela autora, a tarefa de REM que realizou era mais semelhante à tarefa proposta na MUC do que no HAREM. Contudo, esse factor tem pouca relevância na arquitectura do sistema, pois são os exemplos anotados usados para treino que condicionam o tipo de regras aprendidas pelo algoritmo de co-treino.

[2] A diferença entre anotador e classificador reside sobretudo no facto de que no primeiro caso o sistema tem por tarefa delimitar e classificar num texto as entidades que encontra, sendo avaliado de acordo com medidas de precisão e de abrangência, enquanto um classificador tem por objectivo classificar uma lista previamente identificada de entidades, sendo avaliado de acordo com uma medida de correcção.

aprendidas para classificar entidades em novos textos. Ambas as fases partilham os módulos de identificação (de entidades e contextos envolventes) e extracção de características (em inglês, *features*). A fase de teste contém ainda um módulo de propagação que produz um texto final anotado.

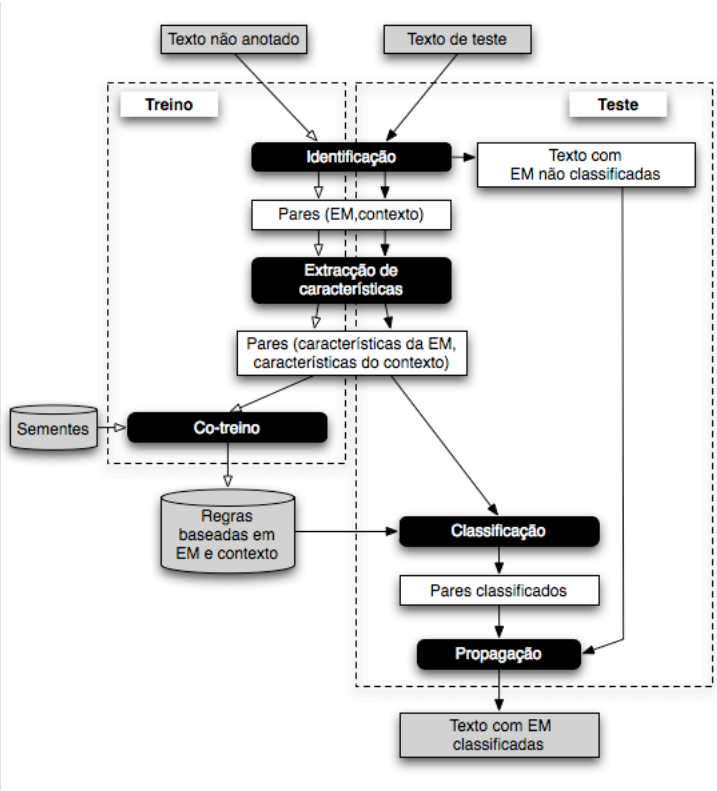


Figura 10.1: Arquitectura geral do sistema R3M

O sistema R3M distingue-se do sistema de Mota (2009) sobretudo ao nível da implementação.

A principal diferença da nossa implementação para a do sistema no qual nos inspirámos é o facto de termos substituído o sistema NooJ (Silberztein, 2004), o qual era usado pelo módulo de identificação, pelo conjunto de ferramentas JET (Grishman, 1999-2006). Este último conjunto de ferramentas apresenta as seguintes vantagens:

- foi concebido a pensar em extracção de informação, mas inclui os vários módulos típicos de processamento de linguagem natural (consulte-se a tabela 10.1, a qual lista os vários módulos e apresenta em destaque os que foram usados nesta fase de desenvolvimento);
- permite criação manual de regras, assim como a sua aprendizagem automática;
- é facilmente portável para outros sistemas operativos, por estar implementado em Java;

Tabela 10.1: Módulos do Jet; as ferramentas marcadas com "X" foram usadas pelo R3M

Módulos do Jet	**Módulos usados pelo R3M**
Atomizador	X
Segmentador de frases	X
Consultador de dicionário	X
Etiquetador morfossintáctico (HMM^3)	X
Etiquetador de EM	-
Analisador de grupos nominais	-
Analisador sintáctico	-
Analisador sintáctico estatístico	-
Reconhecedor de padrões	X
Resolvedor de referências	-

- permite parametrização e criação de pequenos programas de invocação.

Apesar do conjunto de ferramentas Jet ter sido concebido tendo em vista o processamento de textos escritos em inglês, a arquitectura era suficientemente genérica para podermos utilizar os vários módulos no processamento de textos em português. Para isso, seria naturalmente necessário ter dados em português para treinar os módulos que tivessem sido treinados com dados em inglês, ou então criar novas regras no caso em que as regras fossem específicas de inglês.

Quanto aos módulos de classificação e co-treino, originalmente implementados em Lush (Bottou e LeCun, 2003), estes módulos foram reimplementado em R (R Development Core Team, 2008), com o objectivo de no futuro poder vir a tirar partido dos vários módulos de análise estatística existentes no R.

Em seguida descreveremos sucintamente cada módulo e os recursos envolvidos.

10.1.1 Identificação

Quer estejamos numa fase de treino ou de teste, o módulo de identificação é responsável por identificar, em textos não anotados, candidatos a entidades e o contexto em que se encontram. O resultado produzido por este módulo é uma lista de pares constituídos por entidade e contexto.

Este módulo tem duas fases principais: detecção de candidatos a EM e detecção dos contextos em que as entidades ocorrem.

10.1.1.1 Detecção de candidatos a EM

O objectivo da primeira fase é detectar os candidatos a EM. A fase de detecção é composta pelos seguintes passos: atomização, delimitação de frases, consulta de dicionários e aplicação de regras que identificam ou excluem candidatos.

As regras de exclusão tentam evitar que certas sequências de palavras iniciadas em maiúscula sejam marcadas como candidatas. Por exemplo, uma vez que não estamos a fazer reconhecimento de expressões temporais, criámos uma regra de exclusão para os nomes dos meses e de estações do ano. Também excluímos palavras que se iniciassem por maiúscula e que reunissem uma das seguintes condições: (i) fossem palavras vazias (em

inglês, *stopwords*) ou (ii) estivessem ligadas à palavra seguinte por um hífen. A lista de palavras vazias, que constituiu um dos poucos dicionários que usámos[4], foi obtida seleccionando todas as palavras gramaticais (cerca de 3500) contidas no Port4NooJ, o módulo de português para o NooJ descrito em Barreiro (2008).

As regras de identificação de candidatos limitam-se a delimitar sequências de palavras iniciadas por maiúscula. Estas sequências podem também incluir um conjunto limitado de elementos de ligação[5], desde que a palavra inicial e final sejam iniciadas por maiúscula. Dado que foi fornecida pela organização do Segundo HAREM uma lista de palavras em minúscula que podiam iniciar uma entidade mencionada (a lista pode ser consultada no apêndice A, secção A.6), essas palavras se existissem no texto também foram incluídas como fazendo parte do candidato a EM. Esta lista constituiu o outro dicionário que usámos e que contém cerca de 170 entradas[6].

10.1.1.2 Detecção do contexto da EM

Nesta fase, candidatos a EM que se encontrem em determinados contextos definidos por um pequeno conjunto de regras são identificados juntamente com o respectivo contexto. Por contexto deve entender-se uma sequência de palavras que ocorra antes ou depois do candidato a EM. Os pares de candidato a EM e contexto serão fornecidos ao módulo de classificação.

Salientamos que simplificámos os contextos de Mota (2009), de forma a que não necessitássemos de um analisador sintáctico. Precisámos mesmo assim de informação morfossintáctica e por esse motivo treinámos o etiquetador morfossintáctico do Jet com base nos textos da Floresta Sintá(c)tica (Afonso et al., 2002).

Os contextos que considerámos podem não corresponder a um constituinte sintáctico, pois não impusemos nenhuma estrutura sintáctica em particular à sequência de palavras que constituem o contexto. Apenas definimos as seguintes restrições:

- o limite à esquerda, quando existe, de um contexto à esquerda do candidato a EM deve corresponder a: artigo, palavra vazia, preposição ou sequência de dois atómos separados por hífen ou "/";
- o limite à direita de um contexto à esquerda do candidato a EM deve corresponder a: nome, adjectivo ou forma verbal, seguido ou não de vírgula ou qualquer das palavras permitidas como limite à esquerda de um contexto à esquerda;
- o limite à esquerda de um contexto à direita do candidato a EM deve corresponder a: nome, adjectivo, forma verbal ou *que*, antecedido ou não de vírgula;
- o limite à direita, quando existe, de um contexto à direita do candidato a EM deve corresponder a: palavra vazia, preposição ou artigo.

Estas restrições, muito genéricas, foram obtidas por observação de vários exemplos, e ainda precisam de mais experimentação e refinamento.

[4] Cada entrada do dicionário é constituída por uma palavra vazia associada à etiqueta `stw`.

[5] Como elementos de ligação, considerámos as preposições *de*, *em*, *por* e *para* contraídas (excepto no último caso) ou não com o artigo definido, e também os caracteres "-" e "/".

[6] Neste dicionário, as entradas relativas a cargos têm a etiqueta `crg` e as entradas correspondentes a formas de tratamento têm a etiqueta `ft`.

No caso dos contextos à esquerda, estávamos sobretudo a tentar captar (i) contextos em que a entidade estivesse integrada num grupo nominal ou preposicional, (ii) contextos em que a entidade estivesse aposta a um grupo nominal e (iii) contextos verbais de que a entidade pudesse ser o complemento do verbo. A tabela 10.2 ilustra exemplos de entidades e respectivos contextos à esquerda que são detectados por este módulo.

Tabela 10.2: Pares de entidades e contexto à esquerda

Contexto à esquerda	Entidade
Dividir o	IRA
o aeroporto de	Londres
o sector mais violento do	IRA
O seu fundador,	Michael Collins
o segundo mais sobrecarregado com barracas da	Área Metropolitana de Lisboa
As imagens emocionaram o	País
o momento mais incrível do	Mundial
viajava para	Lisboa

Com os contextos à direita tentámos encontrar (i) grupos nominais apostos às entidades ou (ii) construções verbais em que o sujeito poderia ser a entidade. Na tabela 10.3 ilustram-se algumas entidades com o seu contexto à direita.

Tabela 10.3: Pares de entidades e contexto à direita

Entidade	Contexto à direita
Karl Wendlinger	, piloto da
Paikou	, ficou quase completamente submerso pelas
Hamas	está interessado
Aung San Suu Kyi	continua presa
David Bernardino	, prestigiado médico

Criámos ainda regras que associam informação contextual às restantes entidades envolvidas numa estrutura de coordenação de entidades, quando a primeira ou a última das entidades coordenadas tenham sido previamente associadas a informação contextual. No caso de ser a primeira entidade, associa-se o seu contexto à esquerda e, de forma simétrica, no caso de ser a última entidade associa-se o seu contexto à direita. No exemplo 10.1, o contexto à esquerda de *Guiné* vai ser igualmente o contexto de *Angola* e *Moçambique*, enquanto no exemplo 10.2, o contexto à direita de *Foca* vai ser também o contexto de *FIA*.

(10.1) Os novos governos da *Guiné*, de *Angola* e de *Moçambique*

(10.2) *FIA* e a *Foca* omitiram essa informação

Além disso, nos casos em que entidades sejam seguidas de outras dentro de parêntesis, a informação de contexto de uma é associada à outra. No exemplo 10.3, é a informação do contexto à direita de *AR-Santana* que é associada a *Administração Regional de Santana*; no exemplo 10.4 é o contexto à esquerda de *IML* que é associado à entidade dentro de parêntesis.

(10.3) *Administração Regional de Santana* (*AR-Santana*) culpam o

(10.4) o laudo do *IML* (*Instituto Médico Legal*)

Uma vez que era nosso objectivo minimizar a dependência de textos manualmente anotados (neste caso, a Floresta não foi manualmente anotada, mas foi manualmente revista), uma das nossas ideias futuras era limitar o contexto à esquerda e à direita de outro modo, por exemplo, considerar como relevante uma janela de n palavras, ou até que o próximo canditato a EM seja encontrado, em vez de obrigar as palavras limite a serem de uma determinada categoria morfossintáctica.

10.1.2 Extracção de características

O módulo de extracção de características analisa a lista de pares entidade-contexto e cria uma nova lista constituída por pares de vectores de características. Um dos vectores tem as características próprias da entidade, e o outro vector tem as características referentes ao contexto.

Como características da entidade considerámos: a entidade em si, cada constituinte individualmente (excepto elementos de ligação), se a entidade só tem letras maiúsculas e o comprimento da entidade (as entidades com mais de cinco constituintes ficam todas com o mesmo comprimento, seis); como características do contexto usámos: o contexto completo, cada constituinte do contexto e o tipo de contexto (se é à esquerda ou à direita). Em ambos os casos, as palavras vazias não são consideradas constituintes individuais.

Por exemplo, considerando a entidade *Paikou* cujo contexto à direita é *, ficou quase completamente submerso pelas* (ver tabela 10.3), obtém-se o seguinte par de vectores:

```
((entidade=Paikou, inclui=Paikou, sigla=falso, comprimento=1),
 (contexto=, ficou quase completamente submerso pelas,
 inclui=ficou, inclui=submerso, tipo=direito))
```

10.1.3 Classificação

Este módulo determina a classificação dos pares de vectores de características obtidos pelo módulo de extracção de características. Para tal, o módulo usa um conjunto de regras (de classificação) que são inferidas por um algoritmo de co-treino, como explicado na secção 10.1.4.

Uma regra (de classificação) corresponde a um triplo (x,y,z) em que z, designada *precisão* da regra, corresponde a uma estimativa da probabilidade condicional $p(y|x)$ de observar a categoria y quando a entidade tem a característica x. As características tanto podem ser referentes à entidade em si como ao seu contexto.

A classificação de uma entidade (representada por um par de vectores de características) é escolhida usando a regra que tiver maior valor de precisão de entre o conjunto de regras aplicáveis a essa entidade. O conjunto de regras aplicáveis é constituído por todas as regras cuja característica x faça parte do vector de características da entidade.

Como já referimos, cingimos o leque de categorias possíveis às categorias `PESSOA`, `ORGANIZACAO` e `LOCAL`. Adicionalmente, usámos uma categoria extra, `OUTRA`. Esta categoria

não existe no conjunto de categorias da avaliação, e é utilizada para dar conta de entidades que não pertencem a nenhuma das categorias que nos interessavam, mas que podiam ter sido extraídas pelos módulos anteriores.

10.1.4 Co-treino

Tal como descrevemos na secção anterior, as regras de classificação são triplos (*x*,*y*,*z*), em que x é uma característica, y a categoria associada à característica e z a precisão da regra. Estas regras são inferidas incrementalmente de forma semi-supervisionada, usando um algoritmo de co-treino. Este algoritmo parte de um pequeno conjunto de regras e aprende novas regras a partir de pares entidade-contexto não classificados.

O primeiro algoritmo de co-treino foi proposto por Blum e Mitchell (1998) para classificar páginas da rede. A ideia central é aprender alternadamente regras sobre duas vistas diferentes, mas complementares, que se tem sobre um determinado problema a partir de um pequeno conjunto de exemplos classificados e de uma grande quantidade de dados não classificados. No caso da classificação de entidadade mencionadas, tal como proposto por Collins e Singer (1999), uma das vistas é a própria entidade e a outra vista é o contexto em que ela se encontra (o algoritmo 10.1 descreve os passos envolvidos na aprendizagem baseada em co-treino).

Algoritmo 10.1: Algoritmo de co-treino implementado no sistema R3M

Require: S /* Sementes constituídas por características internas classificadas /*
Require: N /* Pares não classificados, (em_i, c_i), em que $em_i = (em_{i1}, \ldots, em_{im})$ é o vector de características extraídas da EM i e $c_i = (c_{i1}, \ldots, c_{1n})$ é o vector de características extraídas do contexto da EM i /*
1: C /* Pares classificados, (em_i, c_i) /*
2: regras_EM /* Regras baseadas em características extraídas da EM /*
3: regras_contexto /* Regras baseadas em características extraídas do contexto da EM /*
4: $n \leftarrow 5$
5: $p \leftarrow 0.95$
6: $\alpha \leftarrow 0.1$
7: regras_EM $\leftarrow$ S
8: **while** $n < 2500$ **do**
9: $C \leftarrow$ `Classificar`(N, *regras_EM*)
10: *regras_EM* $\leftarrow$ `Aprender`(*entidades*(C), α, n, p)
11: $C \leftarrow$ `Classificar`(N, *regras_contexto*)
12: *regras_EM* $\leftarrow S \cup$ `Aprender`(*contextos*(C), α, n, p)
13: $n \leftarrow n+5$
14: **end while**
15: $C \leftarrow$ `Classificar`(N, *regras_EM* $\cup$ *regras_contexto*)
16: *regras_finais* $\leftarrow$ `Aprender`(C, α)

A primeira vista que o algoritmo usa é a das entidades, ou seja, o primeiro conjunto de regras a ser aplicado, designadas "*sementes*", contém regras de classificação que dizem respeito a características extraídas de entidades. Estas regras são aplicadas aos pares entidade-contexto não classificados que foram extraídos nos passos anteriores. Por exemplo, se

o conjunto de sementes fosse constituído apenas pela regra (entidade=Paikou, LOCAL, 0,95), todos os pares entidade-contexto cuja entidade fosse *Paikou* seriam classificados como LOCAL.

Em seguida, o algoritmo infere regras de contexto, com base nos contextos dos pares que forem classificados nesse primeiro passo (ou seja, usa a vista do contexto para obter novas regras). Por exemplo, se as características do contexto de um dos pares classificados fosse (contexto= ficou quase completamente submerso pelas,inclui=ficou,inclui=submerso,tipo=direito), seria possível gerar uma regra por cada característica de contexto desta entidade, em que *y* seria a categoria com que o par foi classificado (no caso, LOCAL) e *z* seria a precisão estimada dados todos os pares classificados pelo algoritmo.

As regras de contexto inferidas são usadas para classificar novamente os pares entidades-contexto. A partir dos novos pares classificados, o algoritmo pode agora inferir regras baseadas nas características das entidades.

O passo de classificação usa o mesmo método de classificação descrito na secção 10.1.3. Em cada passo de inferência de regras, apenas as *n* regras mais frequentes por categoria e que tenham uma precisão acima de um certo limiar são adicionadas ao novo conjunto de regras.

As sementes usadas pelo algoritmo de co-treino foram obtidas a partir da colecção dourada do Primeiro HAREM. Para cada entidade classificada como PESSOA, ORGANIZACAO ou LOCAL criámos uma regra (*x*,*y*,*z*) em que *x* é a característica entidade= preenchida com a entidade que ocorre na colecção dourada, *y* é a categoria mais frequente para essa entidade na colecção dourada e *z* é a probabilidade da entidade ter essa categoria estimada a partir da colecção dourada.

Como só estávamos interessados em pessoas, organizações e locais, todas as entidades da colecção dourada que não pertencessem a essa categoria foram passadas para a categoria OUTRA, excepto entidades TEMPO e VALOR que foram ignoradas. Desta forma poderíamos treinar o sistema com quatro categorias, em que uma delas representa exemplos negativos, em vez de treinar só com as três em que estávamos interessados.

Os pares entidade-contexto não classificados utilizados pelo algoritmo foram extraídos da colecção do Primeiro HAREM, de acordo com os passos ilustrados na fase de treino da figura 10.1 (ver secção 10.1); a colecção dourada do Mini-HAREM foi usada como colecção de teste durante a fase de desenvolvimento do sistema.

10.1.5 Propagação

Este módulo só é aplicado se estivermos numa fase de teste, de modo a produzir a anotação final do texto. A sua função é reconhecer entidades que não se encontram nos contextos representados nas regras descritas na secção 10.1.1.2, mas que podem ser idênticas a entidades que já foram reconhecidas nas fases anteriores e que têm uma classificação associada.

Tomemos como exemplo a EM *Portugal* nas frases 10.5 e 10.6.

(10.5) De regresso ao reino de *Portugal*, «mais cheio de glórias que de despojos», foi bem acolhido por D. Manuel

(10.6) Em *Portugal*, o Instituto Nacional de Saúde elaborou cenários de uma eventual pandemia de gripe humana de origem em aves

No primeiro caso (10.5), *Portugal* encontra-se num contexto que é capturado pelas regras de contexto: *De regresso ao reino de*, formando-se assim um par entidade-contexto que será fornecido ao algoritmo de classificação; porém, no segundo caso, *Portugal* não se encontra num contexto previsto pelas regras e, portanto, essa ocorrência não vai ser analisada pelo modo de classificação.

O módulo de propagação vai então analisar essa ocorrência como uma entidade cuja classificação será a classificação que mais vezes é produzida para *Portugal* na fase de classificação.

Caso *Portugal* ocorra integrado noutro candidato a entidade, essa ocorrência será ignorada pelo módulo de propagação. Por exemplo, na frase 10.7, *Portugal* encontra-se integrado numa entidade maior, *Artes Tradicionais de Portugal*, que também não foi reconhecida num contexto previsto nas regras. De forma a não segmentar essa entidade (e dado que optámos por não produzir anotações com ALT), essa ocorrência de *Portugal* não é tida em conta.

(10.7) foi aberta a exposição internacional «*Artes Tradicionais de Portugal*»

Essencialmente, este módulo é utilizado para aumentar a abrangência do sistema, uma vez que permite a classificação de entidades que não foram classificadas pelo módulo de classificação, por falta de contexto. No entanto, como o módulo de propagação se limita a escolher a classificação mais frequente, não tendo em conta mais nenhuma informação, a classificação pode não ser a correcta, o que poderá fazer diminuir a precisão.

Veja-se, por exemplo, que *Portugal* em 10.8 deverá ser classificado de forma diferente (`PESSOA|ORGANIZACAO`) do que em 10.6 (`LOCAL`). No entanto, o módulo de propagação atribui a ambos a mesma classificação.

(10.8) Quando Granada caiu e a reconquista cristã se impôs então a toda a Península, os dois reinos católicos, *Portugal* e Espanha

As entidades classificadas pelo módulo de classificação como `OUTRA` são ou ignoradas por este módulo ou anotadas apenas como entidades, sem conter os atributos de classificação. Neste último caso, a ausência de classificação tem o significado de que o sistema identifica a sequência delimitada como uma entidade e que a sua classificação não é nenhuma das três que queríamos analisar.

10.2 Resultados

Devido a problemas no módulo de aprendizagem das regras de classificação que não foram resolvidos atempadamente, os resultados da nossa participação no Segundo HAREM acabaram por ficar reduzidos à identificação de entidades mencionadas.

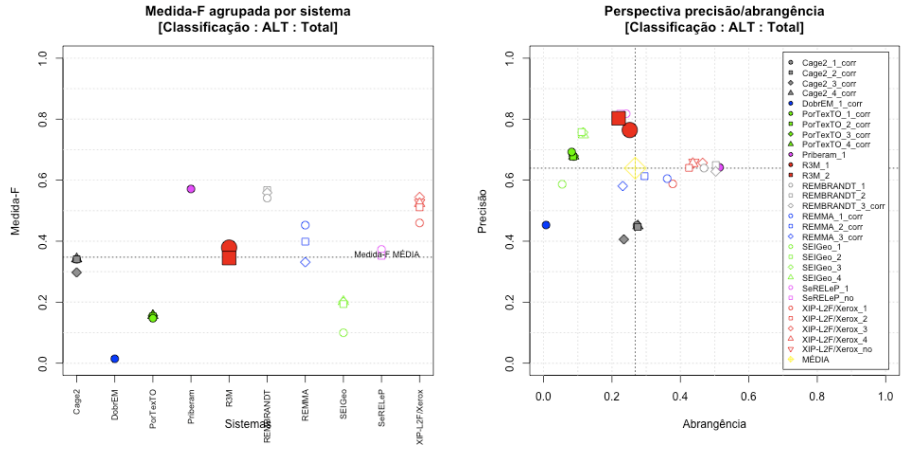
A nossa ideia inicial era participar com duas corridas. Uma corrida incluiria todas as entidades classificadas como `PESSOA`, `ORGANIZACAO` e `LOCAL`, e também as entidades classificadas como `OUTRA` (neste último caso, a anotação não incluiria classificação, de modo a indicar que o sistema as identificou como entidades, mas não as conseguiu classificar como pertencendo a uma das três classes que pretendia reconhecer). A outra corrida incluiria apenas as entidades classificadas como `PESSOA`, `ORGANIZACAO` e `LOCAL`, o que quer dizer que as entidades classificadas como `OUTRA` seriam descartadas antes de produzir o resultado final.

Pretendíamos deste modo verificar se era preferível manter as entidades no resultado final, mesmo que não se soubesse a sua classificação, participando num cenário mais ambicioso (todas as categorias menos `VALOR` e `TEMPO`), ou anotar apenas `PESSOA`, `ORGANIZACAO` e `LOCAL`, participando apenas num cenário selectivo com essas três categorias.

Como os problemas que ocorreram levaram a que não pudéssemos distinguir as entidades que seriam classificadas pelo algoritmo como `OUTRA` das restantes (pessoas, organizações e locais), pois o algoritmo começou a associar a todas as entidades a mesma categoria, acabámos por participar com duas corridas num cenário selectivo com todas as categorias menos `VALOR` e `TEMPO` (cenário selectivo 3):

- R3M_1, que inclui todas as entidades que são identificadas, mesmo as que não se encontram em contextos previstos pelas regras de detecção de contextos;
- R3M_2, que inclui apenas as entidades que são identificadas em contextos previstos pelas regras de detecção de contextos, e ainda as entidades que são reconhecidas pelo módulo de propagação.

Apesar de não termos feito classificação, começamos por mostrar na figura 10.2 o desempenho obtido pelas nossas corridas na classificação das entidades no cenário total com avaliação estrita de ALT, que corresponde ao cenário ideal que os sistemas deveriam alcançar.



(a) Medida F (b) Precisão e abrangência

Figura 10.2: Resultados de classificação no cenário total com avaliação estrita de ALT

Embora os resultados não sejam naturalmente os desejáveis (a melhor corrida, R3M_1, obteve 0,3790 de medida F, ficando em 12º lugar), são equiparáveis aos de outros sistemas que fizeram classificação, como o REMMA, ou ainda melhores, como em comparação com o Cage2. Em particular, na figura 10.2(b) vê-se que o ponto mais forte do sistema R3M em relação a estes dois sistemas é ter um dos melhores valores de precisão que ronda os 0,8, para valores relativamente semelhantes de abrangência (apenas uma das corridas

Tabela 10.4: Identificação no cenário selectivo 3 (todas as categorias excepto TEMPO e VALOR)

Saída	Posição (em 25)	Precisão	Abrangência	Medida F
R3M_1	3	0.7768	0.8134	0.7947
R3M_2	5	0.8116	0.7064	0.7553

do REMMA se destaca com um valor mais elevado). Tal como discutido no capítulo 6, isso mostra que a identificação tem um peso talvez demasiado grande em relação ao da classificação na medida de avaliação.

Centrar-nos-emos, agora, na avaliação da identificação, pois interessa-nos sobretudo perceber o desempenho do sistema na detecção de entidades. Se as entidades não estiverem a ser bem identificadas e delimitadas o algoritmo de aprendizagem estará a treinar sobre dados com mais ruído. Como não utilizámos a etiqueta ALT, mostraremos apenas resultados obtidos com avaliação relaxada de ALT.

Como se pode ver na tabela 10.4, que mostra os resultados obtidos no cenário selectivo 3, a corrida R3M_1 ficou em terceiro lugar na identificação com uma medida F de 0,7947 enquanto a corrida R3M_2 ficou em quinto lugar com 0,7553 de medida F. Estes valores confirmam que apesar de o sistema R3M não ter feito classificação, teve um bom desempenho na identificação das entidades que se propôs reconhecer.

Também se pode ver, e como seria de esperar, que a corrida R3M_1 tem maior abrangência (cerca de 0,11 a mais) do que a corrida R3M_2, pois inclui todas as entidades identificadas na fase de detecção, independentemente do contexto em que ocorrem. Mesmo assim, a precisão dessa corrida é apenas ligeiramente menor (cerca de 0,04) do que a da corrida R3M_2, o que mostra que pode não haver grande vantagem em descartar entidades no caso de não se saber a sua classificação (que é o que a corrida R3M_2 pretende simular: entidades que tenham sido identificadas na fase de detecção de entidades são eliminadas quando se verifica que não ocorrem em pelo menos um contexto previsto pelas regras). Por exemplo, na frase 10.9, *Hugo Estenssoro* e *Londres* são inicialmente identificadas como entidades, cujos contextos não estão previstos nas regras de contexto. Como não existe nenhuma ocorrência de *Hugo Estenssoro* num contexto que possa ser usado para a classificar, esta entidade não fez parte da corrida R3M_2, apesar de fazer parte da corrida R3M_1; *Londres*, como ocorre noutros contextos previstos nas regras, por aplicação do módulo de propagação acabaria por ser reconhecida (e está então anotada em ambas as corridas).

(10.9) *Hugo Estenssoro*, em *Londres*

10.3 Comentários finais

Quando participámos no Primeiro HAREM com o sistema Stencil/NooJ (Mota e Silberztein, 2007), adaptámos um sistema que estávamos na altura a desenvolver para anotar semi-manualmente o CETEMPúblico (Rocha e Santos, 2000) com entidades mencionadas (Mota, 2006). Contudo, essa adaptação não foi total. Em particular, não seguimos o modelo semântico do HAREM, não tentando anotar de forma distinta, por exemplo, *Portugal* nas frases (1.1) a (1.5), ilustradas no capítulo 1: em todos os casos tentámos atribuir a categoria LOCAL.

No Segundo HAREM, ao contrário do que fizemos no Primeiro, que reconhecemos como incorrecto, optámos por seguir mais fielmente as "regras do jogo". Queremos com isto dizer que tentámos desenvolver um sistema que estivesse conforme ao modelo semântico do HAREM e às directivas de anotação. Se assim não fosse, acreditamos que estariamos a enfraquecer a validade do objectivo principal de uma avaliação conjunta que é comparar o desempenho dos sistemas numa tarefa que é comum a todos os participantes.

O maior sucesso da nossa participação foi termos reutilizado um conjunto de ferramentas genéricas que tinham sido desenvolvidas com vista ao processamento de textos escritos em inglês, e aplicado essas ferramentas conjuntamente com recursos portugueses que existiam ou que tivemos de criar.

Sem contar com o facto de a reimplementação só por si constituir uma melhoria do ponto de vista técnico em relação ao sistema em que nos inspirámos, durante o desenvolvimento do sistema de base, fomos melhorando alguns aspectos em relação a esse sistema. Em particular:

- simplificámos as regras de detecção do contexto de EM, o que passou por dispensar um módulo de análise sintáctica;
- incluímos exemplos negativos na fase de aprendizagem, o que evitou criar regras manuais na fase de detecção para excluir entidades de categorias que não queremos reconhecer (o que de certa forma obrigaria a ter praticamente regras para as reconhecer de forma a ter um elevado grau de sucesso na sua exclusão).

Como trabalho futuro gostaríamos de explorar três questões que acábamos por não ter oportunidade de implementar:

- Integrar um módulo de selecção de textos antes da fase de treino, cujo objectivo seria seleccionar textos que pudessem potenciar o resultado do classificador num conjunto de teste. Por exemplo, Ji e Grishman (2006) mostraram que seleccionando frases anotadas mais relevantes é mais importante do que aumentar simplesmente o número de frases do conjunto de treino.
- Detectar o contexto sem necessitar de ter informações morfossintácticas.
- Usar contextos (anotados) também como sementes. Dado que o modelo semântico do HAREM depende fortemente do contexto, estamos em crer que seria mais importante usar contexto como sementes do que entidades classificadas.

Agradecimentos

Agradeço os comentários valiosos e construtivos de Diana Santos, Luís Costa, Bruno Martins, Cláudia Freitas, Hugo Oliveira e Paula Carvalho, em diversas fases de redacção do capítulo, que contribuíram para a sua melhoria substancial.