

# 11

## Conclusões e Trabalhos Futuros

### 11.1

#### Conclusões

Nesta dissertação foi construído um *framework* para geração de sistemas multi-agentes para busca e classificação de documentos textuais de um domínio específico. Foi construída uma infra-estrutura que procurou separar ao máximo as questões referentes à busca e seleção dos documentos das referentes ao algoritmo de classificação utilizado (uma aplicação direta do conceito de *separation of concerns*). Dessa forma, é possível não só acoplar algoritmos já existentes, mas também gerar novos algoritmos levando em consideração características específicas do domínio de documentos abordado. Como as questões estão dissociadas, esta abordagem facilita o processo de seleção do algoritmo, permitindo a reutilização de código e o mínimo de re-projeto.

Mais especificamente, esta dissertação propôs que o processo de geração de um classificador de documentos se dê em duas etapas:

- Uma que lide com questões como a estrutura dos documentos, seu formato e suas especificidades, e também com questões operacionais como detecção de duplicatas, descoberta de novos documentos a serem classificados, seleção de documentos a serem classificados, geração de relatórios, escalabilidade e confiabilidade;
- Uma que lide com questões relacionadas ao esforço de classificação dos documentos, que está diretamente relacionado ao algoritmo de classificação utilizado.

A adoção dessa estratégia de geração possibilita que a segunda parte seja abordada por uma pessoa com perfil diferente: alguém que seja um especialista no domínio dos documentos e que terá condições de desenvolver algoritmos mais

específicos, ou melhorar algoritmos já existentes, através de sua experiência já adquirida. Além disso, como essa parte está dissociada, é possível realizar diversos testes com diferentes algoritmos ou mesmo variações de um mesmo algoritmo, com pouco impacto em re-escrita de código e re-projeto.

Apesar de ter sido adotada uma metodologia de desenvolvimento orientada a objetos para o *design* do sistema, foi utilizada a abstração de agentes de software no nível da compreensão do problema abordado. Essa decisão foi devida ao fato de ser muito mais fácil explicar o sistema em termos de agentes, papéis e tarefas do que em termos de objetos e funcionalidades. Mesmo na descrição do *design* do sistema foram adotadas as abstrações de agentes de software.

A maioria das questões referentes ao sistema multi-agentes, como coordenação entre os agentes, sincronização durante a realização da pesquisa e infra-estrutura para a cooperação formam o *frozen spot* do *framework*.

O outro objetivo desse trabalho foi o de deixar sistema multi-agentes tão escondido quanto fosse possível da segunda etapa do processo de instanciação do *framework*, e também reduzir ao máximo o número de pontos de contato com a primeira etapa. Logo ao se iniciar o trabalho, já foi possível perceber que seria impossível tornar a primeira etapa de instanciação totalmente independente do sistema multi-agentes, pois o grau de inteligência do agente era um dos *hot spots* propostos. Ainda assim, o grande desafio era o de minimizar os pontos de contato com a primeira etapa e não deixar nenhum ponto com a segunda parte.

Estes objetivos foram atingidos, sendo que o único ponto de contato existente foi o previsto e, ainda assim, de forma simples: a instanciação dessa parte se resume à escrita de alguns poucos métodos em duas classes somente, sendo que nenhum deles possui complexidade sequer comparável à complexidade de programação e projeto das questões envolvidas no desenvolvimento de um sistema multi-agente (essa afirmação tem como base as instanciações realizadas), as quais foram tratadas no *frozen spot*. Além disso, o único *hot spot* que pode ser diretamente relacionado ao sistema multi-agentes é a parte da inteligência. Os demais, *hot spots*, apesar de também serem direta ou indiretamente referentes ao sistema multi-agentes, não precisam e nem devem ser encarados dessa forma, pois

estão em um nível de abstração mais baixo – por exemplo, ao instanciarmos o subsistema armazenador de documentos a visitar, estamos na verdade instanciando parte da comunicação entre os agentes, mas não é necessário que essa informação seja levada em conta pelo instanciador. Outro fator importante a ser considerado é que o alto fator de reuso das instâncias desse sistema e também da parte de memória não-volátil do subsistema de memória (ambos pontos de comunicação entre os agentes) foram tão intensos que poderia se pensar em adicioná-los ao *frozen spot* do *framework*, o que caracterizaria ainda mais a transparência do sistema multi-agentes. Se considerarmos apenas a segunda parte da instanciação, relacionada com o algoritmo de classificação, podemos afirmar que o sistema multi-agentes encontra-se totalmente escondido do instanciador.

Não foi objetivo deste trabalho propor uma metodologia para geração de *frameworks* baseados em sistemas multi-agentes onde as questões referentes ao sistema multi-agentes fiquem escondidas no processo de instanciação. Porém, uma conclusão que se pode tirar é que, *quanto mais as questões referentes ao sistema multi-agentes forem independentes da finalidade do sistema final (questões como coordenação, cooperação, comunicação e sincronização), mais essa separação é possível*. Alguns outros trabalhos já realizados propuseram uma abordagem semelhante [71]. Foram geradas quatro instâncias do *framework*, que serviram para exemplificar possíveis usos:

- Com o *Webclipper*, foi possível exemplificar a geração de uma aplicação completa, acabada, escalável e utilizando todas as funcionalidades que o *framework* disponibiliza. Foi também possível exemplificar como a troca do algoritmo de pesquisa pode ser feita de forma bastante direta e independente do resto do sistema. É importante notar que os algoritmos utilizados, apesar de serem suficientes para a aplicação, são bastante simples em complexidade e não utilizavam outros parâmetros do documento além do seu próprio texto, o que vem de encontro com uma das propostas desse trabalho: a complexidade e nível de utilização das meta-informações existentes nos documentos classificados nos algoritmos podem ser tão grande quanto necessário, quem vai definir isso é o domínio dos documentos, no caso, um domínio mais amplo.

- Com o *KM Probe*, foi possível exemplificar a geração de um componente a ser acoplado a uma aplicação maior, além da adoção de um algoritmo mais complexo, baseado em hierarquias de classes. Esse algoritmo tem caráter mais específico que o do *Webclipper*, uma vez que requer uma taxonomia bem definida para obter uma boa margem de sucesso. Foi possível, também, mostrar como se dá o processamento simultâneo de mais de um tipo de documento.
- Com o *Site Seeker*, foi possível exemplificar o uso do *framework* para gerar motores de busca para documentos baseados na *web*.
- Com o *Semantic Probe*, foi possível mostrar como a *web semântica* pode ter papel fundamental para a área de classificação de documentos ao fornecer mecanismos para que os autores agreguem uma quantidade maior de meta-informações aos seus trabalhos.

## 11.2 Trabalhos Futuros

Como continuidade do trabalho realizado no decorrer desta dissertação, sugerimos:

- Testar a utilização de outros algoritmos de classificação, de forma a aumentar a confiabilidade do *framework* quanto a realmente atender a todo esse domínio, em especial tentar avaliar como o *framework* se comporta com algoritmos capazes de aprender com os documentos classificados e com algoritmos capazes de interação com usuários;
- Testar o processamento de outros formatos de documentos textuais;
- Trocar os algoritmos utilizados em cada uma das instâncias de forma a validar se o processo de mudança de algoritmo realmente torna-se mais simples;
- Implementar suporte ao processamento de documentos que não somente os textuais;

- Utilizar as arquiteturas de distribuição propostas de forma a validar a escalabilidade do sistema;
- Testar mais formatos propostos para documentos na *web semântica* de forma a estudar como essa nova *web* pode impactar na área de classificação de documentos;
- Gerar uma ferramenta de busca para *websites* com documentos em algum formato para a *web semântica*. Ferramentas de busca serão a primeira grande classe de aplicações que fazem uso desta nova *web* [72].
- Refinar o algoritmo utilizado pela *Semantic Probe*, de forma a não utilizar somente a classificação do autor como final, mas sim como mais um parâmetro a ser usado na classificação;
- Utilizar este trabalho como um estudo de casos para a criação de uma metodologia de desenvolvimento de *frameworks* baseados em sistemas multi-agentes nos quais as questões referentes ao sistema multi-agentes fiquem concentradas nos *frozen spots*.