MetaAnn: Um Gerador de Ferramentas para Anotação de Textos

Tiago Emanuel Infante Missão[1], **Norton Trevisan Roman**[1]

[1]Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
São Paulo, SP – Brazil

{tiago.missao,norton}@usp.br

Abstract. *With the growing use of corpora in the field of Computational Linguistics, so increases the need for tools that are able to annotate these corpora. These tools, however, are generally aimed at the application of very specific annotation schemes, so that different schemes require the development of specific tools. The use of general tools, on the other hand, brings in extra complexity that is undesirable for the task. In this paper, this subject is approached by introducing the prototype of a meta-tool for generating the Java source code of* ad hoc *annotation tools, thereby retaining the advantages of a general tool, but without the cost in complexity that it brings to the task.*

Resumo. *Com o crescimento do uso de corpora no campo da Linguística Computacional, aumenta também a necessidade de haver ferramentas capazes de anotar esses corpora. Essas ferramentas, no entanto, em geral são construídas para a aplicação de esquemas de anotação bem específicos, fazendo com que diferentes esquemas exijam o desenvolvimento de diferentes ferramentas. O uso de ferramentas gerais, por outro lado, traz consigo uma complexidade extra que não é desejável à tarefa. Nesse artigo, essa questão é abordada através da apresentação do protótipo de uma meta-ferramenta capaz de gerar código-fonte Java de ferramentas de anotação* ad hoc*, mantendo assim as vantagens de uma ferramenta geral, mas sem o custo em complexidade que esta traz à tarefa.*

1. Introdução

Parte importante da Linguística Computacional, a anotação de *corpora* ainda é feita de maneira bastante rudimentar, com uma profusão de ferramentas não raro projetadas para a aplicação de esquemas de anotação específicos (*e.g.* [Craggs and Wood 2004, Maeda et al. 2008, Varasai et al. 2008, Andreas et al. 2012]). Nesse cenário, se algum outro pesquisador desejar, por exemplo, aplicar seu próprio esquema de anotação a um determinado *corpus* de estudo, terá que construir sua própria ferramenta, ainda que esta seja bastante semelhante às ferramentas já existentes para esse mesmo *corpus*, mas que aplicam esquemas de anotação diferentes.

Essa necessidade, por sua vez, resulta em uma considerável quantia de tempo e recursos, tanto humanos quanto financeiros, sendo devotados à construção dessa ferramenta, tempo e recursos estes que poderiam muito bem ser empregados em outras áreas da pesquisa. Como alternativa, o pesquisador poderia, por exemplo, utilizar algum programa de anotação para uso geral (*e.g.* [O'Donnell 2008, Verhagen 2010]). Tal generalização, no entanto, vem ao preço de uma redução na facilidade de uso por parte do usuário final

Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology, pages 11–20,
Fortaleza, CE, Brazil, October 21–23, 2013.

(*i.e.* o anotador), devido aos desvios de foco durante o processo de anotação propriamente dito (em que, por exemplo, o anotador deve tanto segmentar o texto quanto classificá-lo, como em [O'Donnell 2008], ou deve lidar com uma estrutura de camadas abstrata, como em [Verhagen 2010]), além da adaptação da nomenclatura utilizada, necessários para se adequar a ferramenta às necessidades específicas da tarefa.

Consequentemente, o uso de sistemas mais gerais acarreta um aumento da carga cognitiva necessária para a execução da tarefa, potencialmente restringindo seu uso a um público-alvo mais específico, possuidor de conhecimento técnico especializado, e dificultando assim experimentos que envolvam um público mais abrangente. Por outro lado, uma maneira de evitar esse efeito colateral seria passar essa carga ao pesquisador, fornecendo a este um *workbench* sobre o qual ele pode construir mais facilmente seu programa de anotação específico (*e.g.* [Day et al. 2004]). Dessa forma, em vez de uma ferramenta de uso geral, o anotador lidaria com uma ferramenta com menor carga cognitiva, desenvolvida pelo pesquisador.

Embora interessante, essa alternativa tem como desvantagem a necessidade de programação, ainda que de um subconjunto das atividades necessárias à tarefa de anotação, por parte do pesquisador (ou de algum associado), levando novamente ao dispêndio de recursos que poderiam ser melhor aplicados na pesquisa. O desejável, nesse caso, seria um meio de se construir rapidamente ferramentas destinadas a uma única tarefa de anotação, reduzindo assim tanto a complexidade de seu uso quanto o custo de sua produção. Uma maneira de se atingir esse objetivo seria, por exemplo, a construção de uma meta-ferramenta capaz de automaticamente gerar código-fonte de ferramentas de anotação mais simples. Dessa forma, para fazer uso de um novo esquema de anotação, bastaria o pesquisador definir seus detalhes na interface da meta-ferramenta, sem a necessidade de programação, portanto. Esta, por sua vez, produziria um programa de anotação para uso do esquema definido, programa este que poderia então ser usado por um público menos especializado.

Motivada por esses fatores, e projetada para ser usada por não especialistas, a ferramenta aqui descrita – MetaAnn – atinge esses objetivos ao (i) fornecer uma interface gráfica a partir da qual o pesquisador pode definir categorias a serem aplicadas na anotação de um determinado *corpus*, sem a necessidade de desenvolvimento de código; e (ii) gerar uma ferramenta, escrita em Java, de visual "limpo", de modo a não exigir de seus usuários (*i.e.* os anotadores do *corpus*) conhecimento além do necessário para a aplicação do esquema de anotação definido. Embora ainda em sua versão beta, a meta-ferramenta aqui descrita já foi testada na aplicação do esquema de anotação e *corpus* definidos em [Roman and Carvalho 2010] . Distribuída sob a GPL, essa ferramenta está disponível no endereço http://www.each.usp.br/norton/resdial/. Por fim, ainda que tenha sido primariamente testada no *corpus* em questão, nada impede que a ferramenta seja aplicada a outros conjuntos de dados, provido que estes conjuntos estejam codificados segundo o mesmo modelo.

O restante desse trabalho está organizado como segue. Na Seção 2 é feita uma descrição mais detalhada do sistema, bem como sua forma de uso, enquanto que alguns exemplos de saída são encontrados na Seção 3. Na Seção 4, por sua vez, é feita uma comparação entre o sistema aqui descrito e alguns já existentes, apontando as principais diferenças entre eles. Por fim, a Seção 5 apresenta as conclusões e futuras direções para

este trabalho.

2. Descrição do Sistema

O MetaAnn parte do princípio de que o pesquisador já possui um *corpus* de textos codificado conforme o padrão definido em [Roman 2012], segmentado em unidades básicas de anotação[1]. Vale ressaltar que, para o sistema, basta que o *corpus* esteja segmentado conforme esse padrão, não importando o significado dessa segmentação (ou seja, se ela se dá no nível sintático, léxico, semântico ou discursivo). O padrão seguido, por sua vez, corresponde a uma série de *tags* XML a serem aplicadas tanto ao texto em sua forma bruta quanto à sua segmentação e posterior anotação. Ainda que não siga estritamente o padrão de representação LAF[2] (ISO TC37/SC4) [Ide and Romary 2006], o padrão usado pelo MetaAnn é facilmente mapeado a este, uma vez que segue os mesmos princípios, a saber[3]:

- Isolamento do texto em estado bruto (armazenado em XML), sendo que anotações a este são adicionadas de modo *stand-off*, ou seja, via documentos separados ligados a este texto bruto por informação contida em cabeçalhos dentro de cada documento;
- Separação dos formatos usados pelo texto bruto e anotação. Nesse caso, muito embora o MetaAnn use o mesmo formato para ambos, nada impede que futuras versões criem ferramentas que sigam formatos diferentes para a anotação; e
- Separação da estrutura e conteúdo no texto bruto, ou seja, apenas cabeçalhos podem ser associados a ele, além de que deve haver a existência de um documento que defina a segmentação desse texto (*i.e.* suas unidades básicas de anotação).

Dentro do MetaAnn, esquemas de anotação são definidos como uma série de categorias a serem aplicadas a cada unidade básica de anotação do *corpus* de estudo. Assim, para definir um esquema de anotação, o pesquisador deve definir cada uma dessas categorias e seus respectivos valores. Para tal, ele deve acionar o botão *Adicionar* na interface principal do MetaAnn (Figura 1), visualizando então a interface para definição de uma categoria específica (Figura 2). Dentro da interface principal, também é possível ao pesquisador editar e remover as categorias já definidas, além de visualizar informações sobre alguma categoria de sua escolha. Na atual versão do MetaAnn, são apresentados o tipo da categoria (visto mais adiante), seu identificador e lista de opções, quando cabível.

Cada categoria, por sua vez, é tida como pertencendo a um de três tipos diferentes, a saber, Opções Mutuamente Exclusivas, Lista Simples e Texto Livre, que definem quais elementos gráficos serão incluídos na interface da ferramenta de anotação gerada, além de seu comportamento nessa interface. Nesse caso, o primeiro tipo (Opções Mutuamente Exclusivas) é usado quando ao anotador deve ser dada apenas uma opção de classificação, escolhida a partir de uma lista pré-definida, sendo a categoria então codificada como uma *combo box*. O segundo tipo, Lista Simples, oferece ao usuário uma maior flexibilidade, não restringindo-o apenas a uma escolha, porém ainda limitando as possíveis escolhas

[1]Essa segmentação pode ser feita através de uma ferramenta descrita em [Rodrigues et al. 2012], que segue o mesmo padrão de codificação descrito em [Roman 2012], potencialmente livrando o pesquisador da tarefa de desenvolver uma ferramenta assim.

[2]*Linguistic Annotation Framework*.

[3]Consulte [Roman 2012] para maiores detalhes e exemplos do padrão de codificação seguido.

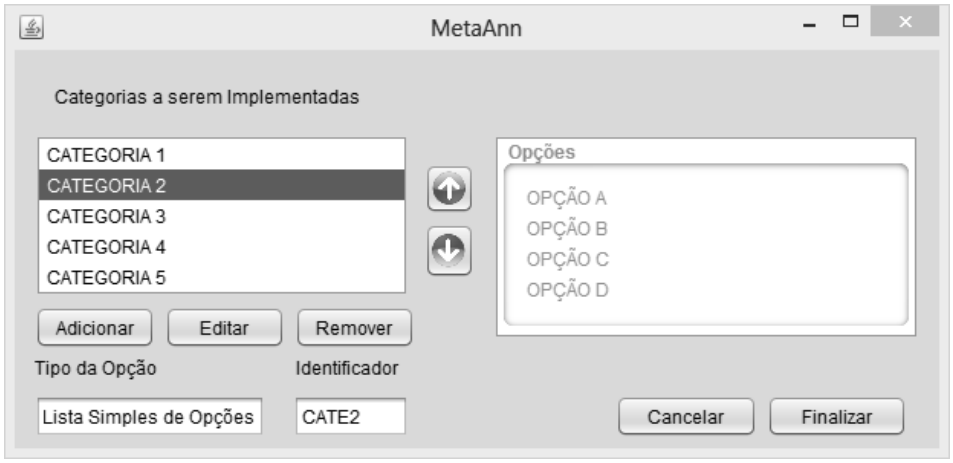

Figura 1. Interface principal do MetaAnn.

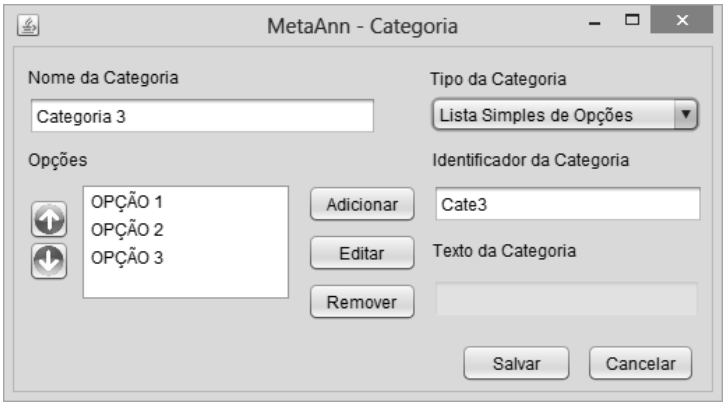

Figura 2. Definição de uma categoria no MetaAnn.

a uma lista pré-definida. Dessa forma, também aqui o elemento gráfico escolhido foi uma *combo box*, porém com a possibilidade de inclusão de múltiplos valores. Por fim, um Texto Livre significa que o anotador deve ter total liberdade para se expressar, sendo então caracterizada por uma *textbox* simples.

Além de definir o tipo ao qual a categoria registrada pertence, o pesquisador também deve fornecer um nome a essa categoria, nome este que constará da interface do programa gerado pelo MetaAnn. De modo a evitar confusões, cada categoria deve possuir um nome distinto das demais registradas. Definido o nome da categoria, o pesquisador deve também criar um identificador único para ela. Tal identificador será usado no código-fonte (Java) em todos os elementos referentes a essa categoria, possibilitando a fácil identificação, dentro do código-fonte, dos elementos envolvidos no funcionamento de cada categoria. Dessa forma, muito embora não seja exigida a edição do código-fonte da ferramenta de anotação, por parte do pesquisador, ela pode ser feita de maneira mais fácil, caso este a julgue necessária.

Caso a categoria a ser definida dependa de uma lista de opções pré-existentes (ou seja, pertença aos tipos Opções Mutualmente Exclusivas ou Lista Simples), o pesquisador deve adicioná-las, uma a uma, através do botão *Adicionar* na Figura 2. Uma janela com um campo textual aparecerá (Figura 3), no qual o pesquisador pode incluir o texto que

será mostrado ao anotador na ferramenta. Esse texto, por sua vez, será repetido no XML resultante da anotação (*i.e.* a saída da ferramenta de anotação), caso o anotador escolha esse valor para a categoria[4]. Se, por outro lado, a categoria definida for do tipo Texto Livre, então o pesquisador pode apenas fornecer um texto inicial a ela, em *Texto da Categoria* (este elemento está desabilitado para os demais tipos). Por fim, terminado o processo de especificação das categorias e opções relevantes, deve-se dar início ao processo de implementação da ferramenta desejada, acionando-se o botão *Finalizar* presente na Figura 1.

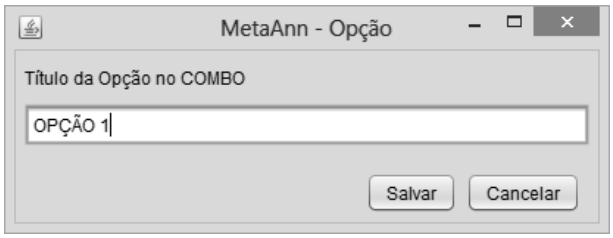

Figura 3. Definição de uma opção dentro de uma categoria.

3. Saída do Sistema

Ao finalizar o processo de definição do esquema de anotação no MetaAnn, o pesquisador terá, como resultado, quatro arquivos-fonte (em um diretório denominado *source*). O primeiro deles, *Arquivo.java*, tem como função controlar a entrada e saída da ferramenta de anotação gerada. O segundo, *Main.java*, tem como principal objetivo fornecer a interface gráfica do programa de anotação. Por fim, *PlainDocument.java* e *Document.java* existem como auxiliares, estabelecendo um canal de comunicação entre *Arquivo.java* e *Main.java*. Além desses arquivos, também é criado um arquivo *Main.jar*, para que possa ser rodado diretamente pelo anotador, bem como o diretório *corpus*, onde o pesquisador deverá disponibilizar o *corpus* a ser anotado e sua segmentação.

Para ilustrar, considere o sistema apresentado na Figura 4[5]. Semelhante ao sistema descrito em [Andreas et al. 2012], sua interface é dividida em contextualização, anotação e navegação[6]. Nesse caso, a parte superior da interface é dedicada à apresentação do corpus ao usuário – sua contextualização. Nela, o anotador tem acesso ao texto do documento em estado bruto, bem como ao identificador e texto da unidade a ser classificada (ou seja, que receberá a anotação). Na parte inferior da interface, por sua vez, o anotador tem acesso a alguns botões para navegação no *corpus*. Na ordem apresentada na figura, são estes: voltar ao início do corpus ($<<$), voltar à unidade de anotação anterior ($<$), ir à próxima unidade ($>$), e ir à última unidade do *corpus* ($>>$). Em seguida há um botão para que o anotador possa salvar o trabalho já feito (em arquivos segundo o padrão definido em [Roman 2012]), além de um campo em que o anotador pode digitar o identificador da unidade de anotação desejada, podendo então ir diretamente a ela. Por fim, há também um botão para deixar a ferramenta, caso em que todo o trabalho feito será também salvo.

[4]Para o futuro, essa etapa deverá permitir tanto a definição do texto para a interface do usuário quanto o texto usado como valor no arquivo XML.

[5]O exemplo apresentado na figura serve apenas para fins ilustrativos, correspondendo a um trecho da obra Dom Casmurro, de Machado de Assis.

[6]Ainda que contextualização e anotação apareçam juntas na interface descrita em [Andreas et al. 2012].

Figura 4. Exemplo de ferramenta gerada, com todos os tipos de categorias possíveis.

Por fim, a parte central da ferramenta é destinada à implementação do esquema de anotação em si, ou seja, a parte definida de acordo com a necessidade e interesse do pesquisador. No exemplo da Figura 4, ela é composta por uma categoria do tipo Opções Mutualmente Exclusivas, seguida por duas do tipo Lista Simples. As duas últimas categorias são do tipo Texto Livre (sendo que na primeira foi definido um texto inicial). Vale notar que, à exceção do texto livre, que é totalmente editável, as demais categorias possuem também um botão *Limpar*, para que o usuário possa remover a classificação já associada a essa categoria. Terminada a anotação, o programa gerará uma série de arquivos XML, contendo a classificação atribuída a cada documento do corpus, com um arquivo para cada documento, conforme definido em [Roman 2012]. Todos esses arquivos são armazenados em um diretório denominado *Annotations*.

4. Trabalhos Relacionados

Dos sistemas existentes para anotação de corpora que fornecem algum meio do pesquisador definir um esquema de anotação, talvez o que mais se aproxime do sistema aqui descrito seja o MMAX2 [Müller and Strube 2006]. Também escrito em Java, o MMAX2 toma por base um texto já segmentado em palavras, ao qual devem ser adicionadas anotações. Essas palavras, por sua vez, podem ser agrupadas em conjuntos contíguos, pelo pesquisador, para formar segmentos de anotação (que, na terminologia do MMAX2 seriam eles mesmos anotações ao *corpus*). A definição dos possíveis atributos e relações entre esses segmentos, feitas pelo pesquisador, constituem no MMAX2 um esquema de anotação, a ser aplicado por seus usuários finais (*i.e.* os anotadores).

Ainda que semelhante ao MetaAnn, o MMAX2 difere deste em que o mesmo sistema usado pelo pesquisador para criar um esquema de anotação é o usado pelo anotador final. Isso faz com que a definição de um esquema de anotação não seja algo tão trivial quanto cliques em uma interface, devendo ser feita por meio de um arquivo XML em

separado. Além disso, toda e qualquer particularidade do esquema fica visível ao usuário final, ainda que por vezes isso não seja desejável (como quando busca-se de maneira indireta medir algum fenômeno). No caso do MetaAnn, além do pesquisador contar com uma interface gráfica para definição do esquema, este fica armazenado no código-fonte da ferramenta gerada, não permitindo acesso direto pelo anotador.

Da mesma forma que no MetaAnn, também no MMAX2 os atributos podem ser tanto do tipo texto livre quanto limitados aos valores de uma lista pré-determinada. A única diferença, contudo, é que no MMAX2 apenas um valor pode ser associado a cada unidade (ou seja, não há correspondente ao tipo Lista Simples do MetaAnn). Isso pode tornar-se um problema quando a unidade contiver mais de uma classificação plausível, de acordo com o esquema adotado, como é o caso de identificação textual de emoção (*e.g.* [Craggs and Wood 2004, Roman and Carvalho 2010]), em que uma mesma unidade pode conter a descrição de emoções distintas, às quais devem ser associados rótulos distintos. Nesse caso, o pesquisador teria que trabalhar com combinações de valores, aumentando assim o número de opções ao usuário, e potencialmente reduzindo a confiabilidade do resultado da anotação, conforme apontado em [Craggs and Wood 2004, Bayerl and Paul 2011].

Também desenvolvido em Java, outro sistema relacionado ao MetaAnn é o Callisto [Day et al. 2004]. Projetado para servir mais como um *workbench* que como uma ferramenta final, o Callisto permite a compilação e instalação de módulos para anotação, por meio de uma biblioteca de funcionalidades (denominadas "serviços de anotação"). Nesse caso, muito embora facilite o desenvolvimento de sistema de anotação *ad hoc*, o Callisto tem como grande desvantagem a necessidade de programação, por parte do pesquisador, do módulo responsável pela aplicação do esquema de anotação ao *corpus*. No MetaAnn isso é feito sem essa necessidade, ainda que, se desejar, o pesquisador possa modificar o código-fonte resultante.

Afastando-se da linha de sistemas desenvolvidos para rodar isoladamente em um computador, há aqueles projetados para uso, tanto por anotadores quanto por pesquisadores, via *web*. O primeiro deles, Brandeis Annotation Tool (BAT) [Verhagen 2010], modela cada tarefa de anotação como uma série de camadas, em que cada camada pode se referir a outras, sendo a primeira geralmente a segmentação do texto em estado bruto. Embora a ideia por trás desse enfoque seja de que, ao separar a tarefa em sub-tarefas (camadas), está-se de fato simplificando-a, a abstração necessária para tal pode gerar uma carga cognitiva extra no anotador, carga esta potencialmente danosa ao resultado geral da anotação. No MetaAnn, a anotação de segmentos de um texto consiste de uma etapa distinta das demais, executada pela ferramenta *ad hoc* gerada, fazendo assim com que o anotador esteja inteiramente focado na aplicação do esquema de anotação em si.

Semelhante ao MetaAnn, também no BAT o pesquisador pode definir a interface apresentada ao anotador, sendo que suas escolhas determinam os elementos gráficos da interface de anotação. Contudo, e do mesmo modo que no MMAX2, também BAT está limitado a um único valor por categoria, enquanto que o MetaAnn abre para a inclusão de múltiplos valores. Além disso, a necessidade de conexão com a internet pode ser um problema, especialmente para anotadores que disponham de pouco tempo extra para a execução da tarefa, sem mencionar que outro problema relatado com esse tipo de abor-

dagem trata das inúmeras diferenças entre *browsers* [Verhagen 2010], o que faz com que o sistema possa não rodar em alguns deles.

Por fim, também dentro da linha de sistemas via *web*, e possuindo também os mesmos problemas de exigência de uma conexão de rede ativa, está o GATE Teamware [Bontcheva et al. 2013]. Da mesma forma que Callisto, MMAX2 e o próprio MetaAnn, esse sistema, através da definição de certos atributos pelo pesquisador, permite a adaptação da interface apresentada ao anotador. A diferença está em que, além desses papéis, o GATE Teamware possui o papel adicional de avaliador – uma pessoa responsável por resolver discordâncias entre anotadores. Além disso, esse é o único dos sistemas citados a possuir suporte ao padrão ISO TC37/SC4 que, embora não adotado pelo MetaAnn, é compatível ao seguido por ele.

Dentro desse sistema, contudo, esquemas de anotação são definidos por arquivos XML (de forma semelhante ao MMAX2), carregados pelo pesquisador. Tais esquemas, por sua vez, são responsáveis por especificar os tipos de atributos e anotações permitidos (*i.e. tags* válidos a serem aplicados ao *corpus*). Nesse ponto, a definição de um esquema de anotação torna-se algo não tão simples, exigindo algum conhecimento por parte do pesquisador de como esse arquivo XML deve ser montado. No MetaAnn, por outro lado, a definição é feita sem que o pesquisador tome ciência de como o esquema é codificado internamente.

5. Conclusão

Neste artigo foi apresentado o MetaAnn – uma meta-ferramenta para geração de ferramentas de anotação *ad hoc*, a partir de atributos definidos em sua interface principal. Em comparação às principais alternativas existentes, o MetaAnn tem como vantagens (a) o isolamento completo do sistema para anotação final, na forma de um utilitário java (e portanto multiplataforma); (b) a apresentação de uma interface gráfica, a partir da qual o pesquisador pode definir o esquema de anotação a ser seguido (através da definição de atributos, seus tipos e valores permitidos), sem a necessidade de criação à parte de arquivos XML ou mesmo de programação; e (c) a possibilidade de uso da ferramenta final, por parte dos anotadores, no momento que melhor lhes convier, sem a dependência de servidores centrais ou conexões de rede ativas.

Como trabalhos futuros, pretende-se permitir a inclusão de dependência entre categorias, como no MMAX2, em que uma delas se tornaria "ativa" somente quando um determinado subconjunto de valores, pertencente a outra categoria, fosse escolhido. Também será feita uma separação entre o texto que aparece como valor associado a uma categoria, na interface para o anotador, e seu valor no arquivo de anotação resultante, permitindo assim uma maior flexibilidade para o pesquisador. Por fim, um último recurso a ser implementado é a inclusão, na interface de anotação, de dados constantes no *corpus* que o pesquisador julgar relevantes.

Agradecimentos

Esta pesquisa contou com o apoio do CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, através do programa PibIC da USP.

Referências

Andreas, J., Rosenthal, S., and McKeown, K. (2012). Annotating agreement and disagreement in threaded discussion. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Bayerl, P. S. and Paul, K. I. (2011). What determines inter-coder agreement in manual annotations? a meta-analytic investigation. *Computational Linguistics*, 37(4):699–725.

Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., and Gorrell, G. (2013). Gate teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, pages 1–23.

Craggs, R. and Wood, M. M. (2004). A two dimensional annotation scheme for emotion in dialogue. In *AAAI Spring Symposium: Exploring Attitude and Affect in Text: Theories and Applications*, Stanford, USA. Technical Report SS-04-07.

Day, D., McHenry, C., Kozierok, R., and Riek, L. (2004). Callisto: A configurable annotation workbench. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation. (LREC 2004)*, pages 2073–2076, Lisboa, Portugal.

Ide, N. and Romary, L. (2006). Representing linguistic corpora and their annotations. In *Proceedings of the Fifth Language Resources and Evaluation Conference (LREC 2006)*, Genoa, Italy.

Maeda, K., Lee, H., Medero, S., Medero, J., Parker, R., and Strassel, S. (2008). Annotation tool development for large-scale corpus creation projects at the linguistic data consortium. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco.

Müller, C. and Strube, M. (2006). Multi-level annotation of linguistic data with MMAX2. In Braun, S., Kohn, K., and Mukherjee, J., editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang, Frankfurt a.M., Germany.

O'Donnell, M. (2008). The UAM corpustool: software for corpus annotation and exploration. In *Proceedings of the XXVI Congreso de AESLA*, Almeria, Spain.

Rodrigues, F., Semolini, R., Roman, N. T., and Monteiro, A. M. (2012). Tseg – a text segmenter for corpus annotation. In *Proceedings of the VIII Brazilian Symposium on Information Systems (SBSI 2012)*, volume 1, pages 700–709, São Paulo – SP – Brasil.

Roman, N. T. (2012). Resdial – descrição da codificação (v.1.0). Technical Report 001/2012, PPgSI-EACH-USP, São Paulo, SP – Brazil.

Roman, N. T. and Carvalho, A. M. B. R. (2010). A multi-dimensional annotation scheme for behaviour in dialogues. In Kuri-Morales, A. and Simari, G. R., editors, *Proceedings of the 12th Ibero-American Conference on Artificial Intelligence (IBERAMIA 2010)*, volume 6433 of *Advances in Artificial Intelligence*, pages 386–395, Bahía Blanca, Argentina. Springer. ISBN: 978-3-642-16951-9.

Varasai, P., Pechsiri, C., Sukvari, T., Satayamas, V., and Kawtrakul, A. (2008). Building an annotated corpus for text summarization and question answering. In *Proceed-*

ings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco.

Verhagen, M. (2010). The brandeis annotation tool. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 3638–3643, Valletta, Malta.