

# ALGORITMOS DE APRENDIZADO DE MÁQUINA APLICADOS À BASE DE CONHECIMENTO DE UM SISTEMA DE APRENDIZADO SEM-FIM

Amaury Mario Ribeiro Neto<sup>1</sup>\*, Edimilson Batista dos Santos<sup>1</sup>

<sup>1</sup>Universidade Federal de São João del Rei – UFSJ  
Departamento de Ciência da Computação – DCOMP  
CEP: 36301-360 – São João del-Rei (MG) – Brazil

amauryMariorneto@outlook.com, edimilson.santos@ufsj.edu.br

**Abstract.** *NELL is a never-ending learning system that aims to have computational systems learned in a continuous and incremental way, being also able to use acquired knowledge to improve their own learning. The objective of this work is to use different machine learning techniques to build models that make inferences and generate new facts in order to populate the knowledge base of the NELL system. The generated models were trained with a database built from information from existing relationships in the NELL knowledge base and compared through evaluation metrics that resulted in good prediction and classification values.*

**Resumo.** *O NELL é um sistema de aprendizado sem fim que propõe que sistemas computacionais aprendam de maneira contínua e incremental, sendo ainda, capazes de se utilizar de conhecimento adquirido para aprimorar seu próprio aprendizado. O objetivo deste trabalho é utilizar diferentes técnicas de aprendizado de máquina para a construção de modelos que realizem inferências e gerem novos fatos a fim de popular a base de conhecimento do sistema NELL. Os modelos gerados foram treinados com uma base de dados construída a partir de informações das relações existentes na base de conhecimento do NELL e comparados através de métricas de avaliação que resultaram em bons valores de predição e classificação.*

## 1. Introdução

O aprendizado de máquina (AM) [Mitchell 1997] é um ramo dentro da área de Inteligência Artificial que vem ganhando cada vez mais popularidade. Uma das razões é a grande quantidade de aplicações possíveis, principalmente em problemas envolvendo *big data* (grandes bases de dados). Este fato contribuiu para um grande avanço e progresso nos resultados obtidos por métodos e algoritmos desta área de pesquisa nas últimas décadas. Mesmo assim, não existem, hoje em dia, ainda muitos sistemas computacionais capazes de aprender de maneira cumulativa para sempre. E mais importante, sistemas que se utilizem do conhecimento adquirido ontem para melhorar sua habilidade de aprendizado hoje, num processo contínuo e sem fim.

---

\*The authors would like to thank UFSJ for supporting this project through the notice 004/2019/PROPE/UFSJ.

A principal característica desses sistemas é a utilização dos conhecimentos previamente adquiridos para aprimorar a aprendizagem de novos dados de forma contínua e incremental em um processo sem fim [Mitchell 2015]. O NELL (*Never-Ending Language Learning*) [Carlson 2010] é considerado o primeiro sistema de aprendizado sem fim relatado na literatura. Este sistema foi colocado em operação em janeiro de 2010 e tem sido executado até os dias atuais. Sua tarefa consiste em buscar diversos tipos de conhecimentos na internet, classificando e relacionando palavras em determinados contextos. Desta forma, o NELL é capaz de assimilar as informações buscadas, conectando entre si as crenças com características.

A ontologia predefinida do NELL apresenta uma estrutura padrão que define classes (“Pessoa”, “Cidade”, por exemplo) e relações (moraEm(pessoa, cidade), por exemplo). Estas classes e relações da ontologia são marcadas como itens que devem ser “populados”. “Popular” um item significa, em nosso contexto, extrair conhecimento sobre este item, a partir da Web, e armazenar na ontologia. É também esperado que o sistema NELL seja capaz de expandir esta ontologia inicial sempre que for necessário adquirir novos conhecimentos.

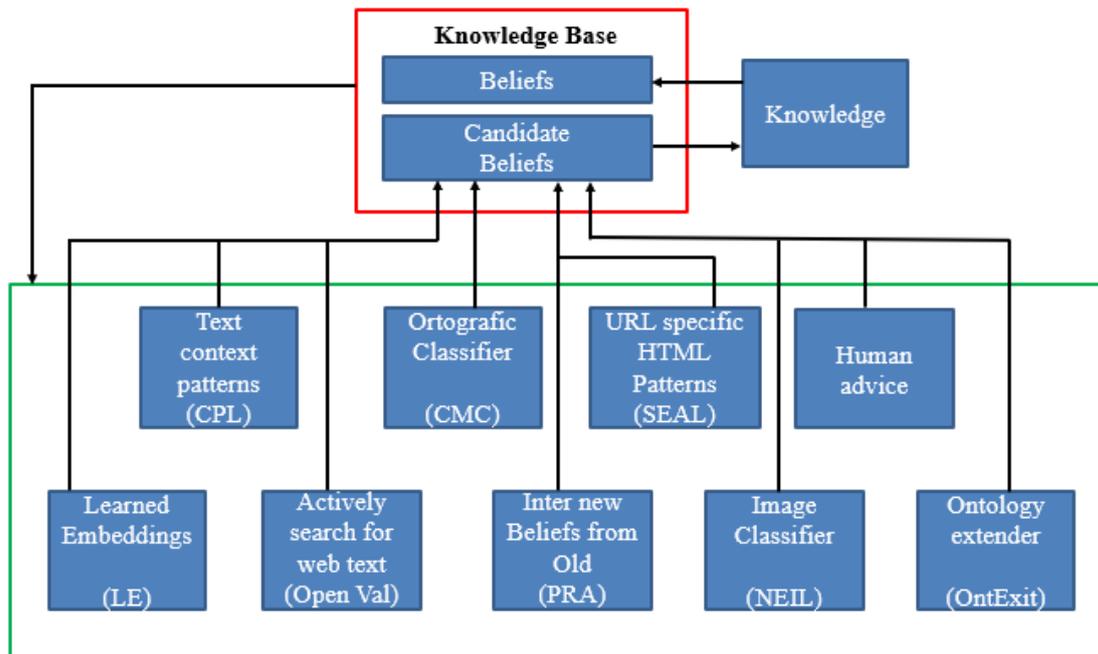
Embora um sistema de aprendizado sem fim não seja algo novo na comunidade de aprendizado de máquina [Thrun and Pratt 1998] [Thrun and Mitchell 1995], desenvolver metodologias para ajudar a estender e “popular” sua base de conhecimento é ainda um desafio. Alguns trabalhos na literatura [dos Santos et al. 2016] [Miani and Junior 2015] [Verma and Hruschka 2012] têm apresentado propostas que ajudam em tais tarefas. Com o intuito de contribuir na tarefa de “popular” a base de conhecimento do sistema NELL, neste artigo, nós propomos a indução de modelos de aprendizado de máquina capazes de realizar inferências e gerar novos fatos.

Diferentes técnicas de aprendizado de máquina foram adotadas e avaliadas, neste trabalho, para a indução dos modelos. Mais especificamente, são apresentados os resultados de 5 algoritmos do estado-da-arte [Peter Norvig 2013]: KNN (*K-Nearest Neighbor*), Árvore de Decisão, Naive Bayes, SVM (máquinas de vetores de suporte) e Redes Neurais de Múltiplas Camadas (MLP). Os modelos foram treinados e testados com um conjunto de dados criado a partir da base de conhecimento do sistema NELL. Este conjunto de dados é constituído por variáveis (ou atributos) e valores aprendidos a partir das relações existentes no NELL. O objetivo principal é verificar se os modelos construídos são capazes de prever novos fatos (ou valores) para serem inseridos à base de conhecimento do NELL, contribuindo assim para o seu aprendizado.

Este artigo é organizado como segue. A Seção 2 apresenta o sistema NELL. Na Seção 3, é descrita a estratégia de aprendizado dos algoritmos a partir dos dados das relações do NELL. A Seção 4 mostra os resultados obtidos pelos algoritmos. Por fim, a Seção 5 exhibe as conclusões e aponta alguns trabalhos futuros.

## **2. *Never-Ending Language Learning (NELL)***

O sistema NELL (*Never-Ending Language Learning*) [Mitchell 2015] foi desenvolvido para, assim como os humanos, adquirir vários tipos de conhecimentos ao longo dos anos. Esse aprendizado será feito de maneira própria e através de um conhecimento prévio adquirido para permitir que o sistema evolua, contando com a supervisão adequada. Se-



**Figura 1. Arquitetura do sistema NELL [Mitchell et al. 2018].**

gundo descrição do site oficial do projeto<sup>1</sup>, desde 2010, o sistema está funcionando continuamente, tentando executar duas tarefas principais por dia:

- Primeiramente, tentativa de “ler” ou extrair fatos de textos encontrados em centenas de milhões de páginas web;
- Posteriormente, ele tenta melhorar sua competência em leitura, para que amanhã possa extrair mais fatos da Web com maior precisão.

O sistema tem como entrada uma ontologia que define centenas de categorias (por exemplo, pessoa, bebida, atleta, esporte) e relações semânticas entre essas categorias. Também é fornecido um conjunto de 10 a 20 exemplos positivos (sementes) para cada categoria e relação, juntamente com uma coleção de 1 bilhão de páginas da web como dados não rotulados, e acesso a 100.000 pesquisas diárias no mecanismo de busca do Google.

O NELL usa um algoritmo de aprendizado multitarefa semi-supervisionado em larga escala que acopla o conjunto de treinamento de mais de 2500 diferentes classificadores e métodos de extração. O sistema é composto por diversos componentes (veja Figura 1) que aprendem de diferentes perspectivas dos dados: por exemplo, uma visão usa características ortográficas de um nome de entidade potencial, e outra usa contextos livres de texto nos quais o sintagma nominal é encontrado. Outra característica relevante é a de ser um sistema que treina a si mesmo com base em sua coleção crescente de crenças confiáveis.

Este sistema de aprendizado contínuo possui um Integrador de Conhecimento (IC) para examinar os fatos candidatos propostos e promove os mais fortemente amparados

<sup>1</sup><http://rtw.ml.cmu.edu>

destes ao status de crença. Em cada iteração, cada subsistema componente realiza seu processamento utilizando como entrada a Base de Conhecimento (BC) atual, e então o IC decide quais fatos candidatos propostos são promovidos. A BC cresce iteração a iteração, provendo mais e mais crenças que são utilizadas por cada subsistema componente para retreinar a si mesmos a aprender a ler melhor na próxima iteração.

Neste sentido, esta abordagem pode ser vista como uma implementação de um método de aprendizado semi-supervisionado acoplado, no qual múltiplos componentes aprendem e compartilham tipos complementares de conhecimento, supervisionados pelo IC. Esta abordagem também pode ser vista como uma aproximação para um algoritmo de Maximização de Expectativa (ME) no qual o passo da Expectativa envolve estimar iterativamente os valores verdadeiros para um conjunto muito grande de crenças candidatas virtuais da BC compartilhada, e o passo de Maximização envolve retreinar os vários métodos de extração dos subsistemas componentes.

Assim, iterativamente, os próprios componentes juntos com o IC concretizam duas importantes propriedades do NELL: auto-supervisão e auto-reflexão. O NELL está funcionando desde janeiro de 2010. Como resultado, possui uma base de conhecimento continuamente crescente com mais de 70 milhões de fatos extraídos (com diferentes graus de certeza), sendo mais de 2 milhões com alto grau de certeza (e alta precisão) considerados crenças. Maiores informações sobre o sistema NELL podem ser encontradas em <http://rtw.ml.cmu.edu>.

### 3. Metodologia

O conjunto de dados construído a partir do conhecimento do sistema NELL e usado para o treinamento e teste dos algoritmos de aprendizado de máquina é composto por uma matriz de 75 colunas. Cada coluna representa uma categoria diferente (um atributo ou variável), com aproximadamente 24 mil linhas, totalizando aproximadamente um milhão e 800 mil valores. As categorias e seus valores foram obtidos a partir de dados das relações armazenadas na base de conhecimento do NELL.

A base de conhecimento do sistema NELL está disponibilizada no site do projeto<sup>2</sup> e contém várias categorias e relações entre elas. Esta base, com milhares de instâncias, contém 13 colunas de dados, porém, apenas as três primeiras colunas foram utilizadas para a criação do conjunto de dados: coluna *Entity*, que armazena nomes de categorias; coluna *Relation*, que contém nomes das relações entre as categorias; e coluna *Value*, que contém valores aprendidos através destas relações. Estas colunas apresentam dados sobre relações entre categorias. O conjunto de dados foi então criado a partir de dados aprendidos pelas relações existentes na base de conhecimento do NELL.

Os dados destas 3 colunas são extraídos da base de conhecimento do sistema NELL. Em seguida, cada relação e seus dados são pesquisados individualmente. Cada relação tem suas variáveis identificadas e separadas em duas colunas, como por exemplo: a relação “companyceo” foi dividida em dois atributos: *company/ceo*, cada um em uma coluna. Os dados que compõem a relação são alocados às colunas, de acordo com os atributos. A Tabela 1 apresenta, como exemplos, 4 dos 75 atributos e seus valores do conjunto de dados criado. Depois de obtidos os atributos e seus valores, algumas etapas

---

<sup>2</sup><http://rtw.ml.cmu.edu/rtw/>

de pré-processamento foram aplicadas antes da aplicação dos algoritmos de aprendizado de máquina.

**Tabela 1. Exemplos de atributos (e seus valores) do conjunto de dados criado a partir do conhecimento do NELL.**

City	Person	Food	Sector
shangai	jacob	potatoes	insurance
atlantic_city	bob	nectar	internet
washington_d.c	john	leaves	marketing
boston	carol_meyrowitz	potato	news
oakland	michael_geoghegan	roots	consulting
atlantic_city	bob	food_value	accounting

Algoritmos de aprendizado utilizam características e estratégias variadas para a classificação dos dados, seja por semelhanças ou relações entre os valores aprendidos durante o treinamento dos modelos. Um exemplo de aprendizado observado foram palavras padrões de uma categoria e até mesmo na própria estrutura das palavras aprendidas pelo NELL. Na categoria ‘person’ por exemplo, nomes de pessoas coletados pelo NELL foram treinados pelo modelo que identifica padrões como nomes comuns: ‘robert’, ‘john’, ‘bob’, ‘sophia’ e também sobrenomes muito usados como: ‘williams’, ‘smith’, ‘miller’. Desta maneira, supondo que o modelo treinado tente classificar o nome: ‘robert\_smith’, por exemplo, sua predição apontaria uma probabilidade alta de pertencer a categoria ‘person’ devido ao nome e sobrenome comum já previamente treinado no modelo, além de se tratar de uma palavra composta.

O estudo dos sufixos e prefixos também revelou padrões que auxiliam na tentativa de predição de valores pelos modelos. Considerando a categoria ‘city’, ao aprender valores como: ‘portland’, ‘oakland’ e ‘cleveland’, o modelo perceberia que palavras com a terminação ‘-land’ teriam altas probabilidades de pertencer a categoria ‘city’ devido ao sufixo. Palavras derivadas de outras já classificadas também possuem grandes chances de pertencer a mesma classe, como por exemplo: ‘potato’ e ‘potatoes’; uma vez que ‘potato’ é classificado como pertencente a coluna ‘food’, a palavra ‘potatoes’ possui alta probabilidade de ser classificada também. Existem ainda outras características usadas para as classificações como: cidades extraídas com terminação em ‘city’ (‘boise\_city’, ‘kansas\_city’), valores da classe ‘sector’ com terminação ‘-ting’ (‘marketing’, ‘accounting’, ‘consulting’), dentre outras.

Além de padrões gramaticais e estruturais, a ocorrência das palavras é outra característica que informa ao modelo em qual classe o dado se enquadra. Esta análise de ocorrência é de difícil percepção humana, porém visíveis aos modelos de inteligência artificial treinados de forma a utilizar análises estatísticas para identificar padrões e melhorar suas taxas de acerto.

### 3.1. Pré-processamento do conjunto de dados

O conjunto de dados criado apresenta dois problemas comuns, devido ao número grande de variáveis. O primeiro problema é referente ao número de valores faltantes (ou ausentes). Muitas das relações semânticas existentes na base do NELL possuem poucas instâncias (valores) e, por isso, muitas variáveis no conjunto de dados criado têm poucas

instâncias e assim apresentam muitos valores ausentes. Algumas delas chegam a ter mais de 80% de valores ausentes. Para tratar este problema, decidiu-se preencher os valores ausentes a partir da replicação dos próprios valores presentes das variáveis.

O segundo problema refere-se ao número de valores das variáveis. Muitas delas apresentam mais de 1000 valores (por exemplo, a variável *company* possui cerca de 5000 valores). Além disso, muitos destes valores têm pouca frequência. Para diminuir o número de valores das variáveis, foi aplicado um filtro implementado no software Weka<sup>3</sup>, chamado *MergeInfrequentNominalValues*. Este filtro mescla todos os valores dos atributos especificados que são insuficientemente frequentes. O filtro foi aplicado em todos os atributos do conjunto de dados, de forma que não tivessem mais que 100 valores cada após a aplicação.

Em seguida, cada valor foi associado a sua respectiva coluna. Para isto, foi realizada a transposta da matriz do conjunto de dados, invertendo suas linhas e colunas afim de que os valores do conjunto sejam corretamente conectados aos seus atributos. Ao final da transformação, cada linha do conjunto de dados é composta por todos os exemplos de seu determinado atributo, considerando-o como uma classe, no formato correto para a execução dos algoritmos de aprendizado de máquina.

As inferências são feitas por meio das relações entre os dados e, através delas, os algoritmos de aprendizado de máquina fazem previsões e podem gerar novos fatos. A escolha das classes a serem preditas pelos algoritmos foram as colunas (categorias) da base de dados, enquanto as demais linhas se tornaram seus respectivos dados. Dessa forma, após o treinamento dos dados, o classificador irá determinar, com uma certa taxa de acurácia, a qual das categorias ou domínios um novo dado irá pertencer (observe a 1).

Uma vez que o conjunto de dados está preparado e organizado, ele é convertido da estrutura de dados da biblioteca python *Pandas* para o tipo *array*. Feito isso, o array do conjunto de dados foi 'achatado' com o método *ravel* para que pudesse ser remodelado pela função *reshape* para o formato desejado. Ambos os métodos são fornecidos pela biblioteca *Numpy*.

O algoritmo *One Hot Encoder* (OHE) foi aplicado para transformar os valores categóricos dos atributos em valores numéricos. Para o treinamento do classificador Naive Bayes (mais rápido do que outros algoritmos), utilizou-se o *Label Encoder* (LE) que, diferente do OHE, não gera sequências binárias.

#### 4. Análise de Resultados

Para os experimentos com o conjunto de dados criado, foram utilizados 5 algoritmos de aprendizado de máquina do estado-da-arte: KNN, SVM, árvore de decisão, Naive Bayes e rede neural de múltiplas camadas (MLP). Estes algoritmos foram treinados usando a estratégia de treinamento (70%) e teste (30%) e também de validação cruzada (ou *cross-validation*, com *k-fold=5*).

Os parâmetros dos algoritmos foram definidos com a ajuda de uma função do *Python* chamada *GridSearchCV*, que realiza uma busca exaustiva pelos melhores valores dos parâmetros a partir de um conjunto de valores fornecido. A Tabela 2 apresenta os

---

<sup>3</sup>O Weka é um programa de código aberto escrito em Java que possui uma coleção de algoritmos de *Machine Learning e Data Mining*, desenvolvido pela Universidade de *Waikato*, Nova Zelândia.

valores dos parâmetros dos algoritmos que trouxeram os melhores resultados. Os demais algoritmos mantiveram seus respectivos parâmetros padrões inalterados.

**Tabela 2. Algoritmos com otimização de parâmetros.**

Algoritmo	Parâmetro(s) Alterado(s)
KNeighborsClassifier (KNN)	$n\_neighbors = 6$
MLPClassifier (Rede Neural)	$activation = 'tanh'$
SVM	$kernel = "rbf", C = 10, gamma = 0.1$

A Tabela 3 apresenta os resultados obtidos por meio da estratégia de treinamento e teste para cada algoritmo de aprendizado de máquina. Cada algoritmo foi executado 10 vezes e a média dos resultados é exibida.

**Tabela 3. Média dos resultados de 10 execuções obtida por cada algoritmo, com diferentes métricas.**

Métricas	KNN	Árvore de Decisão	SVM	Naive Bayes	MLP
Acurácia	0,94518	0,95002	0,95043	0,94990	0,94894
Precisão	0,94900	0,95527	0,95611	0,95522	0,95331
Revocação	0,94518	0,95003	0,95043	0,94990	0,94894
F1-score	0,94435	0,94946	0,94979	0,94929	0,94832
Curva-roc	0,97221	0,97469	0,97491	0,97466	0,97413

A Tabela 4 apresenta os resultados de acurácia dos algoritmos, treinados por validação cruzada ( $k-fold = 5$ ). É possível observar que os valores obtidos por cada algoritmo são muito próximos dos exibidos na Tabela 3 (treinamento e teste), o que diminui o risco de haver *overfitting* (sobreajuste ou superajuste).

**Tabela 4. Resultados obtidos pelos algoritmos, treinados através da estratégia de validação cruzada ( $k-fold = 5$ ).**

Métricas	KNN	Árvore de Decisão	SVM	Naive Bayes	MLP
Acurácia	0.93895	0.95001	0.94993	0.95005	0.94951

Quanto ao tempo de treinamento, observou-se que a árvore de decisão e Naive Bayes são os mais rápidos. O SVM e as redes neurais MLP tiveram melhor desempenho após o refinamento dos parâmetros. Já o KNN apresenta o maior tempo de treinamento entre todos os algoritmos.

## 5. Conclusões

A base de conhecimento do sistema NELL é grande e diversificada, com uma grande variedade de categorias e relações entre elas. Porém, os subsistemas do NELL ainda não são capazes de aprender muito sobre elas. Neste trabalho, é apresentada uma metodologia para ajudar o sistema NELL a aprender novos fatos para preencher (ou "popular") sua base de conhecimento, treinando algoritmos de aprendizado de máquina a partir do próprio conhecimento aprendido pelo NELL.

A estratégia proposta utiliza relações e seus dados da base de conhecimento do NELL para criar um conjunto de dados e treinar algoritmos de aprendizado de máquina.

Diferentes algoritmos foram usados nos experimentos, como Naive Bayes, árvore de decisão, SVM e redes neurais de múltiplas camadas, e os resultados obtidos demonstraram que a estratégia é promissora. Os modelos de classificação induzidos obtiveram taxas de classificação acima de 90%, o que indica que são capazes de aprender sobre diferentes atributos (ou categorias) a partir do conjunto de dados criado e inferir novos fatos para popular a base de conhecimento do NELL.

Os algoritmos Naive Bayes e árvore de decisão tiveram os melhores tempos de treinamento, utilizando o conjunto de dados criado. Para um conjunto de dados maior, os algoritmos SVM, redes neurais e KNN poderão utilizar técnicas de paralelismo e serialização, como o método *Pickle*.

## Referências

- Carlson, A. (2010). Toward na architecture for never-ending language learning. Proc. Of the Twenty-Fourth AAAI Conference on Artificial Intelligence.
- dos Santos, E. B., Fernandes, M. L., Jr., E. R. H., and Duarte, M. C. (2016). Bayesian networks for identifying semantic relations in a never-ending learning system. In Madureira, A. M., Abraham, A., Gamboa, D., and Novais, P., editors, *Intelligent Systems Design and Applications - 16th International Conference on Intelligent Systems Design and Applications (ISDA 2016) held in Porto, Portugal, December 16-18, 2016*, volume 557 of *Advances in Intelligent Systems and Computing*, pages 279–288. Springer.
- Miani, R. G. L. and Junior, E. R. H. (2015). Exploring association rules in a large growing knowledge base.
- Mitchell, T. (2015). Never-ending learning. Proceedings of the Conference on Artificial Intelligence (AAAI).
- Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Yang, B., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., and Welling, J. (2018). Never-ending learning. *Commun. ACM*, 61(5):103–115.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.
- Peter Norvig, S. R. (2013). *Inteligência Artificial*. Elsevier Editora, third edition. ISBN 8535237011, 9788535237016.
- Thrun, S. and Mitchell, T. (1995). Lifelong robot learning. *Robotics and Autonomous Systems*, 15(1):25 – 46.
- Thrun, S. and Pratt, L. (1998). *Learning to Learn: Introduction and Overview*, page 3–17. Kluwer Academic Publishers, USA.
- Verma, S. and Hruschka, E. R. (2012). Coupled bayesian sets algorithm for semi-supervised learning and information extraction. In *ECML/PKDD*.