Ciência de Dados e Big Data

Recuperação da Informação na Web e em Redes Sociais

PUC-Minas IEC | Pós-Graduação Lato Sensu

Zilton Cordeiro Jr.

PUC Minas

professorzilton@gmail.com

www.dcc.ufmg.br/~zilton

Projeto Final

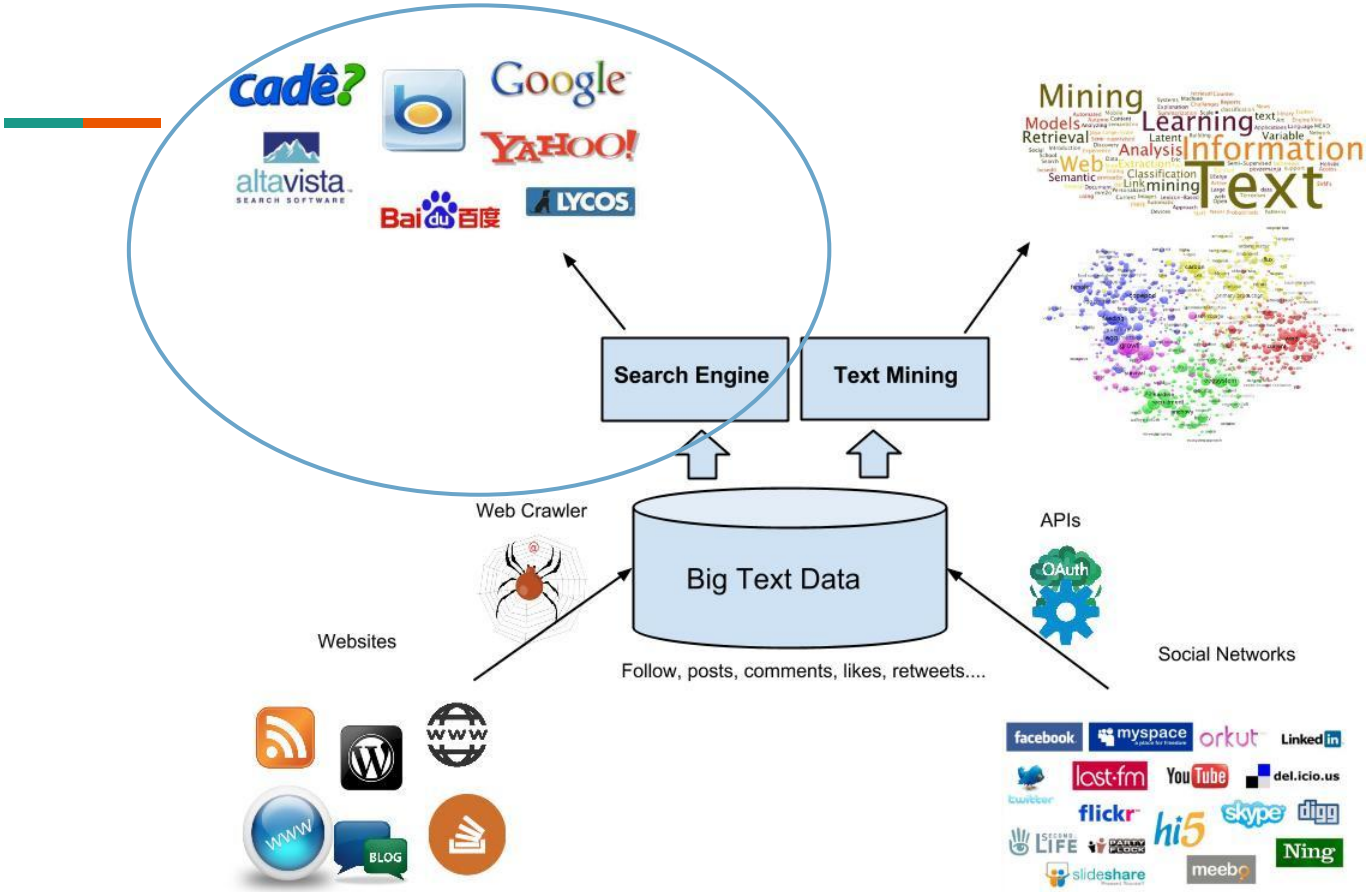
- O **projeto final** consiste em realizar um estudo da Web para um assunto real e de livre escolha.
  - Exemplos: Automóveis, moda, música, imóveis...

- Será necessário
  - Coletar dados em texto de redes sociais e sites da Web
  - Analisar o conteúdo textual obtido
  - Analisar dados de relacionamentos entre usuários (i.e. nas redes)
  - **Relatório final**

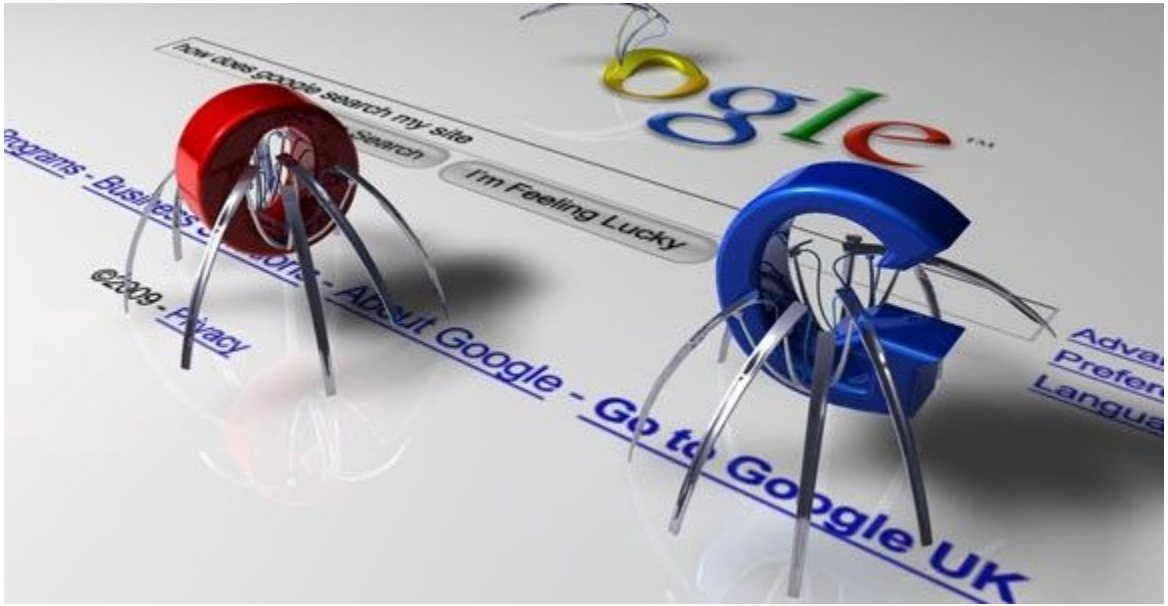
- **Data de Entrega**
  - 15° dia após a última aula às 23:59hrs

Busca Textual e Similaridade

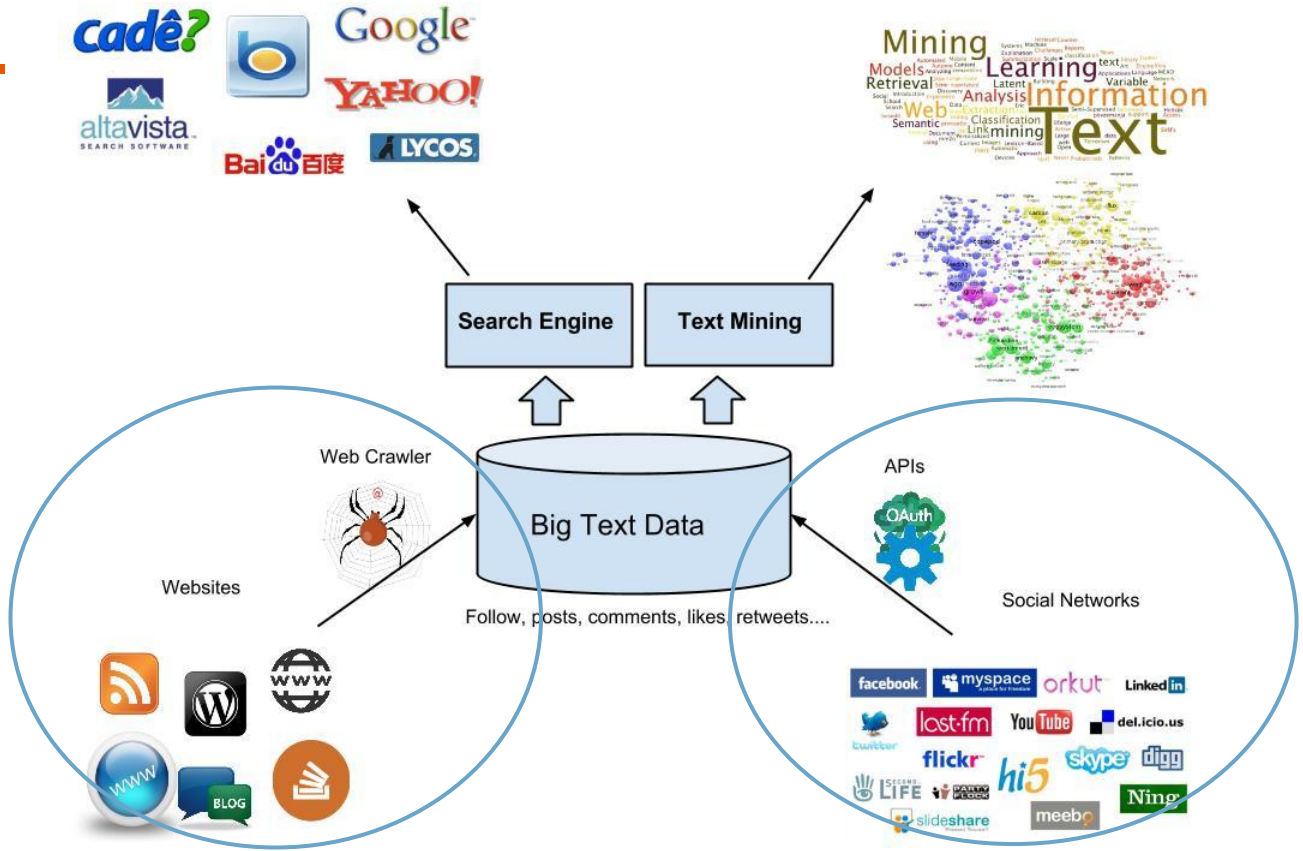
Mineração da Web e Redes Sociais



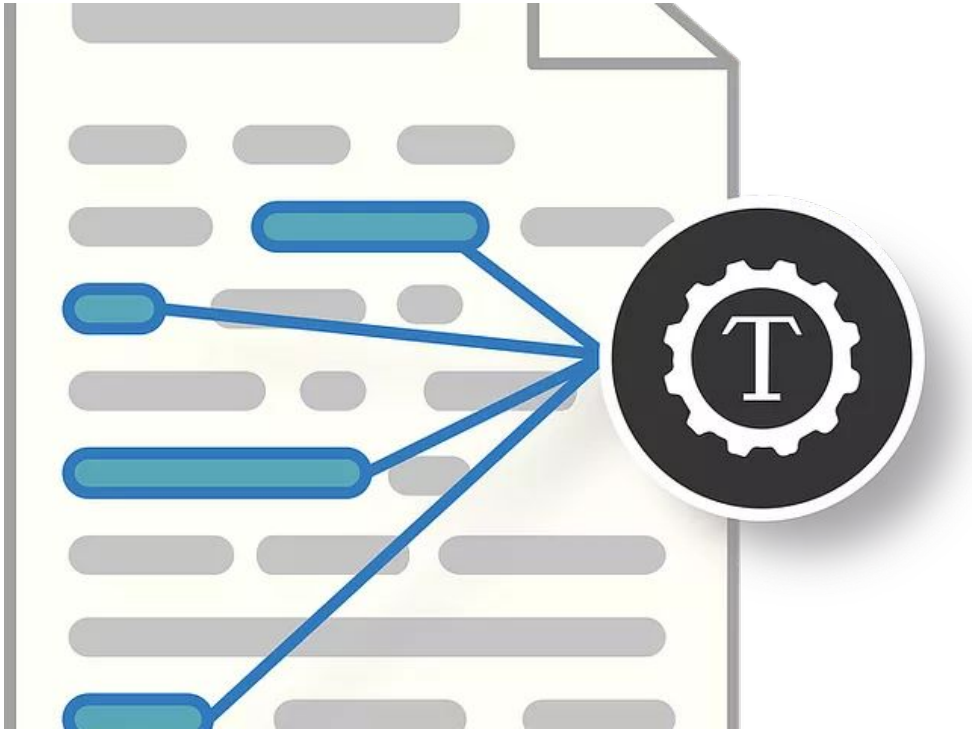
Coleta de Dados



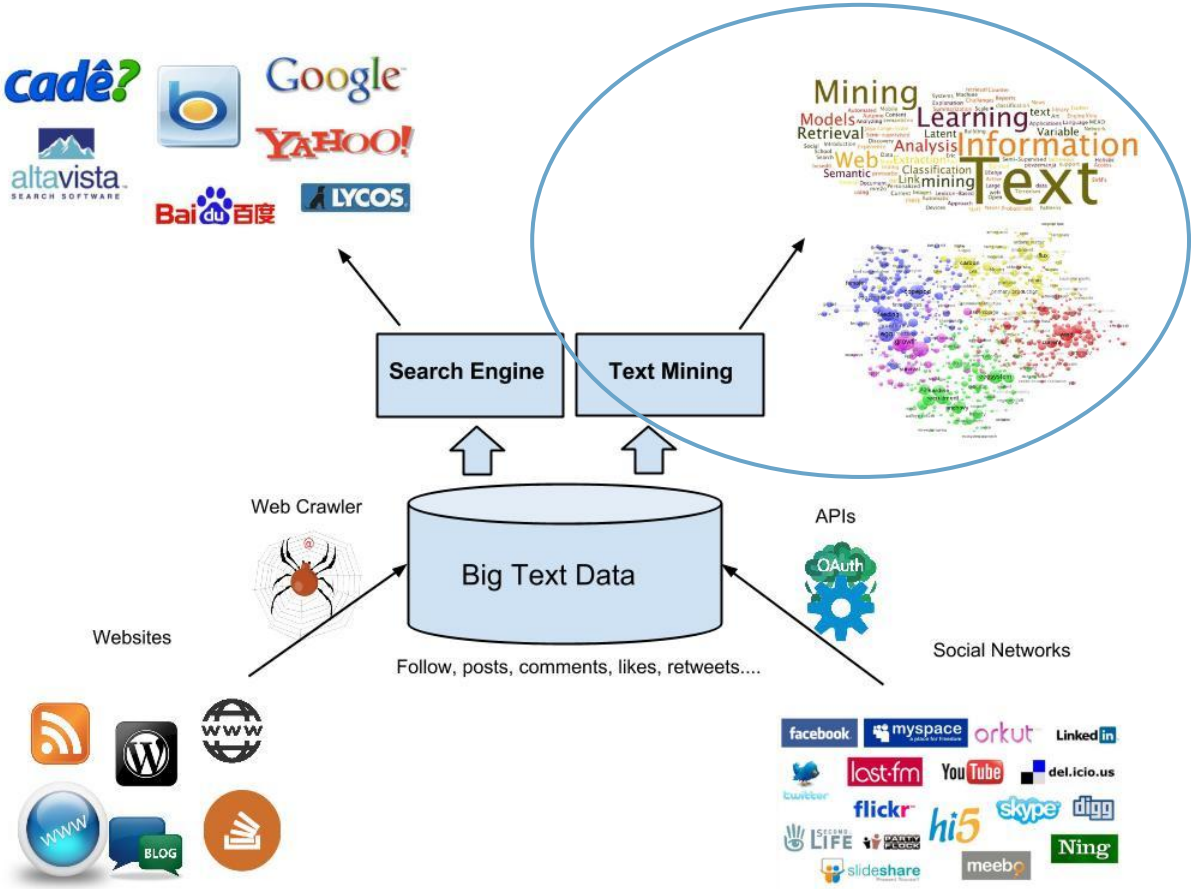
Mineração da Web e Redes Sociais



Mineração de Textos



Mineração da Web e Redes Sociais



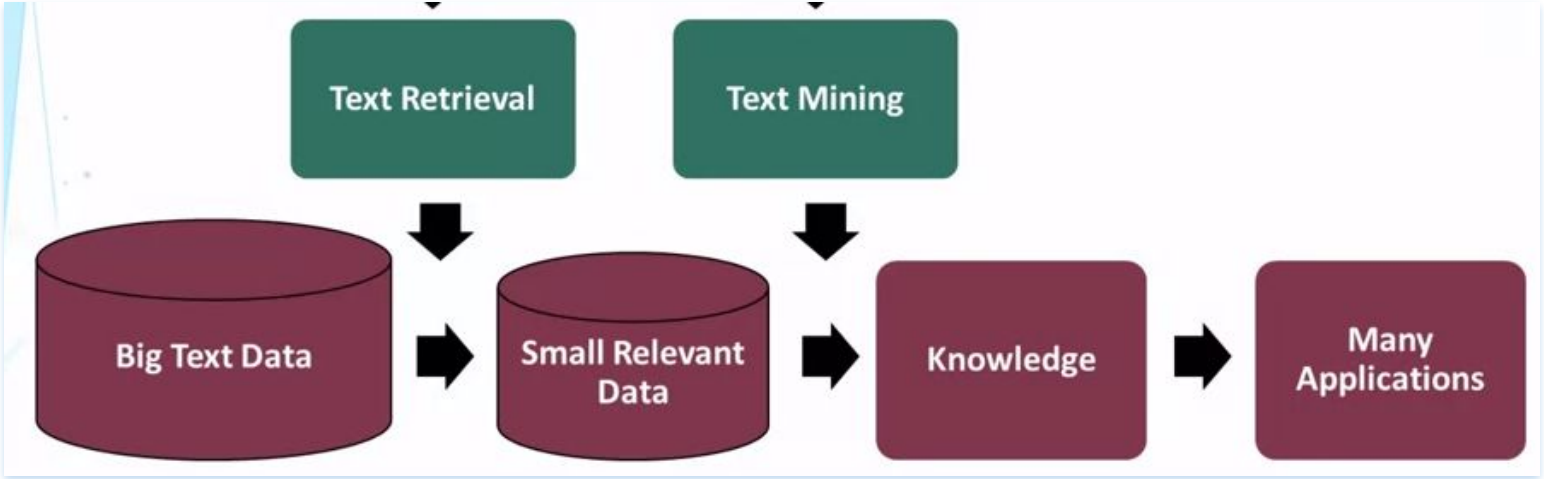
Mineração de Textos

❖ Recuperação de Texto e Mineração

- ➢ Minimizar o esforço humano ao consumir grandes volumes de dados e fornecer conhecimento para tomadas de decisão otimizadas
- ➢ Recuperação de texto é um componente essencial de qualquer sistema de mineração de textos
- ➢ Recuperação de texto pode ser um pré-processador para mineração de textos

Mineração de Textos

- **Recuperação + Mineração**



Processamento de Linguagem Natural (NLP)

- **Segmentação de Palavras (Bag of Words)**

  "O cachorro está perseguindo o garoto no playground"

  - Generalizam menos que a representação por caracteres
    Ex: difícil identificar palavras (Chinês)
  - São as unidades básicas da comunicação humana
  - Possível contagem de palavras mais frequentes
  - Podem ser usadas para formar tópicos a partir de conexão de palavras
  - Se algumas palavras são positivas e outras são negativas é possível realizar análise semântica

Processamento de Linguagem Natural (NLP)

❖ POS - Part of Speech tags

“O	cachorro	está	perseguindo	o	garoto	no	playground“
Det	Noun	Aux	Verb	Det	Noun	Det	Noun

- ➢ Método complementar à representação em palavras
- ➢ Possível contar adjetivos, sujeitos, verbos, sujeitos associados a quais verbos
- ➢ Enriquece a representação do texto

Processamento de Linguagem Natural (NLP)

❖ Detecção de Entidades

"O cachorro está perseguindo o garoto no playground"
Animal — Pessoa — Localização

- ➢ Adição de entidades e relacionamentos
- ➢ Análise semântica das palavras
- ➢ Relações
  - o cachorro estava perseguindo o garoto
  - o garoto está no playground

- ➢ Sujeito mais frequente em uma coleção de artigos e notícias
- ➢ Coocorrência: Pessoas geralmente citadas em conjunto
- ➢ Menos robusto que identificação de palavras ou até análise sintática

Processamento de Linguagem Natural (NLP)

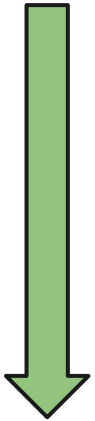
❖ Tradeoff

"O cachorro está perseguindo o garoto no playground" — **String de Caracteres**

"O cachorro está perseguindo o garoto no playground" — **Segmentação em Palavras**

Det Noun Aux Verb Det Noun Det Noun — **POS Tagging**

Animal Pessoa Localização — **Detecção de Entidades**



Mais esforço humano e mais propenso a erros

Mais próximo da representação humana de conhecimento

Processamento de Linguagem Natural (NLP)

❖ Humanos e Algoritmos

- ➢ Computadores não são capazes de obter uma representação de texto completamente correta
- ➢ É preciso combinar a colaboração humana com sistemas computacionais
- ➢ Padrões extraídos do texto podem ser interpretados por humanos e humanos podem fornecer informações e dados anotados que tornam os algoritmos mais efetivos (Algoritmos de classificação)

Processamento de Linguagem Natural (NLP)

❖ Associações entre palavras

- ➢ Associações entre termos para sugerir variações de consultas
- ➢ Construção automática de mapas de tópicos: palavras são vértices e conexões são arestas (grafos na próxima aula!)
- ➢ Comparar e sumarizar opiniões: quais palavras estão mais fortemente associadas a “**bateria**” em *reviews* **positivos** e **negativos** a respeito do **iPhone6**, respectivamente?

Mineração de Tópicos

- **Ou ainda...**
  - Agrupamentos (clustering)
  - Categorização ou Classificação
  - Mineração de regras de associação
  - Trending Topics

KNIME - Enriquecimento

Atribuição de Tags

- Nós de Enriquecimento mudam a granularidade dos termos
- Detecção de entidades, part of speech…
- Para cada entidade detectada (Termo) uma tag é adicionada, especificando seu tipo
- Um termo pode ser associado a mais de uma tag

Estrutura de Termos

❖ Atributos

- ➢ Texto
- ➢ Tags

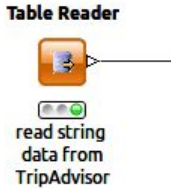
Term	Document
Great[JJ(POS)]	"Great food, intere...
food[NN(POS)]	"Great food, intere...
,[SYM(POS)]	"Great food, intere...
interesting[JJ(POS)]	"Great food, intere...
service[NN(POS)]	"Great food, intere...
this[DT(POS)]	"Great food, intere...
restaurant[NN(POS)]	"Great food, intere...
is[VBZ(POS)]	"Great food, intere...

Fluxo para Prática - Enriquecimento

- **Fluxo**: *text mining-pratica*
- Enriquecer palavras com tags semânticas
- Detectar entidades
- Configurar dicionários de entidades

Enriquecimento (Inglês)

❖ Leitura e Tratamento



Leitura de dados do TripAdvisor em tabela

Read table - 2:2 - Table Reader(read string)

File

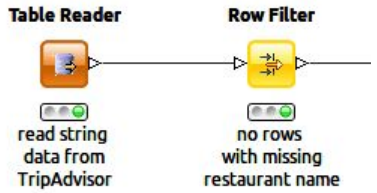
Table "default" - Rows: 449 | Spec - Columns: 7 | Properties | Flow Variables

Row ID	user name	review title	review text
Row0_1_R...	1travellerBruss...	Great food, interesting servi...	this restaurant is seemingly
Row0_2_R...	coverdriven	Excellent Lunch Destination	Very much enjoyed our expe
Row0_3_R...	Lula12783	Hidden treasure near KaDaWe	We found this little gem thro
Row0_4_R...	Deise_Boy08	Excellent Food Very Reason...	Food was top class and very
Row0_5_R...	Tanguyatea	Good food, great prices!	I went there because it was
Row0_6_R...	OlgaNottingham	Nice food at a reasonable p...	I am no expert on Vietnames
Row0_7_R...	Jack D	Good food and entertaining ...	From reading other reviews
Row0_8_R...	Rick M	Very good	A very tasty Vietnamese mea
Row0_9_R...	Alexg123	Excellent food and servive	Great food at very reasonab
Row0_10_...	stabreim	A nice surprise	I found that this restaurant
Row0_1_R...	mooniesss	Ok but not that great.	I followed trip adviser rating

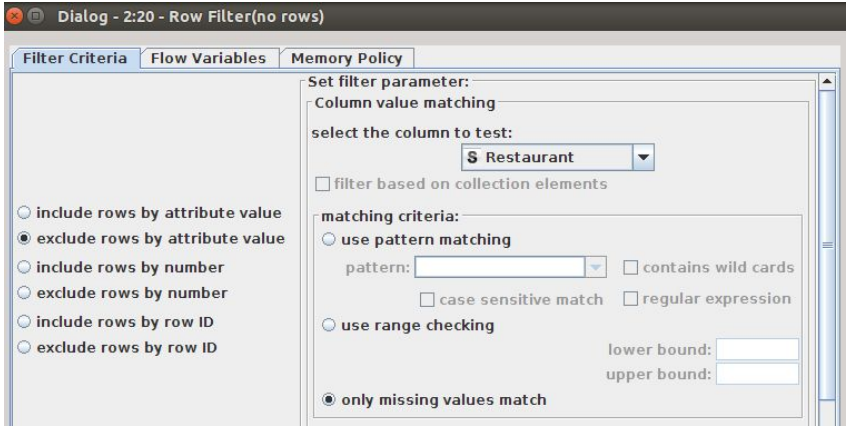
Obs: Oportunidade para revisar a transformação de dados genéricos em documentos

Enriquecimento (Inglês)

❖ Leitura e Tratamento



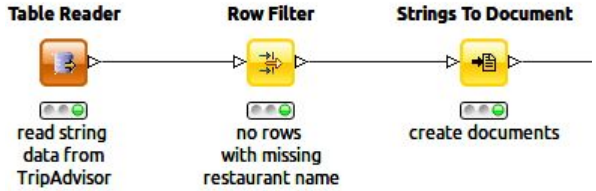
Filtragem de dados incompletos (missing)



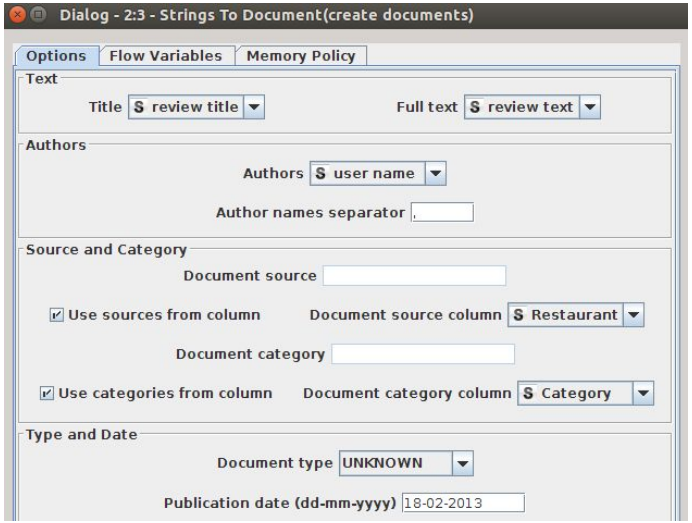
Excluir linhas cuja coluna “Restaurant” não está preenchida.
Opção: “only missing values match”

Enriquecimento (Inglês)

❖ Transformação em Documentos



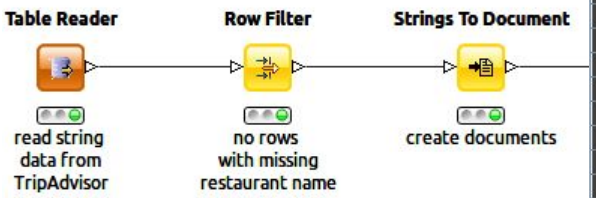
Configure os metadados do documento. Qual coluna você quer usar para representar o título, o texto, autor...



Enriquecimento (Inglês)

❖ Transformação em Documentos

Lembrando que é criada uma coluna chamada Document no fim da tabela original



String and document output table - 0:3 - Strings To Document (cre

File

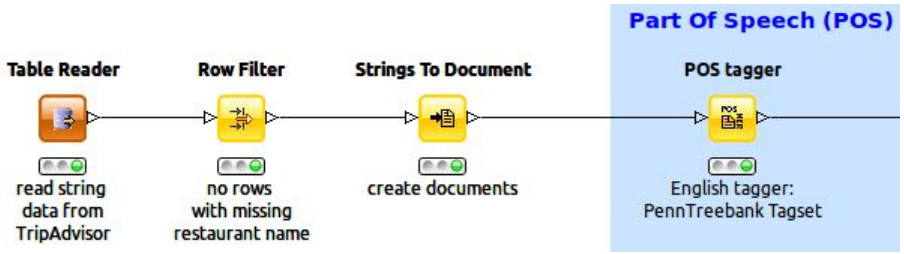
Spec - Columns: 8 | Properties | Flow Variables

Table "default" - Rows: 440

Row ID	Reviews	Restaurant	Categ...	Document
Row0_1_R...		Saigon and mo...	Asian	"Great food, interesting servic...
Row0_2_R...		Saigon and mo...	Asian	"Excellent Lunch Destination"
Row0_3_R...		Saigon and mo...	Asian	"Hidden treasure near KaDaWe"
Row0_4_R...		Saigon and mo...	Asian	"Excellent Food Very Reasona...
Row0_5_R...		Saigon and mo...	Asian	"Good food, great prices!"
Row0_6_R...		Saigon and mo...	Asian	"Nice food at a reasonable pri...
Row0_7_R...		Saigon and mo...	Asian	"Good food and entertaining ...
Row0_8_R...		Saigon and mo...	Asian	"Very good"
Row0_9_R...		Saigon and mo...	Asian	"Excellent food and servive"
Row0_10_...		Saigon and mo...	Asian	"A nice surprise"
Row0_1_R...		Saigon and mo...	Asian	"Ok but not that great."
Row0_2_R...		Saigon and mo...	Asian	"This is.. BY FAR.. my favorite r...
Row0_3_R...		Saigon and mo...	Asian	"Tasty, good looking food. Per...
Row0_4_R...		Saigon and mo...	Asian	"Good food with warm welcom...
Row0_6_R...		Saigon and mo...	Asian	"So happy to have found this j...
Row0_7_R...		Saigon and mo...	Asian	"lived up to the hype"
Row0_8_R...		Saigon and mo...	Asian	"Saigon is in Berlin"
Row0_9_R...		Saigon and mo...	Asian	"Great experience, great food!"
Row0_10_...		Saigon and mo...	Asian	"Great value, good food, inter...
Row0_1_R...		Saigon and mo...	Asian	"Ok food, entertaining owner ...
Row0_2_R...		Saigon and mo...	Asian	"no-BS delicous food and ent...
Row0_3_R...		Saigon and mo...	Asian	"Excellent"
Row0_4_R...		Saigon and mo...	Asian	"Wow!"

Enriquecimento (Inglês)

❖ Part Of Speech (POS-Tagging)



Não exige configuração.

Vai procurar pela coluna **Document**

➢ Nó **POS tagger**

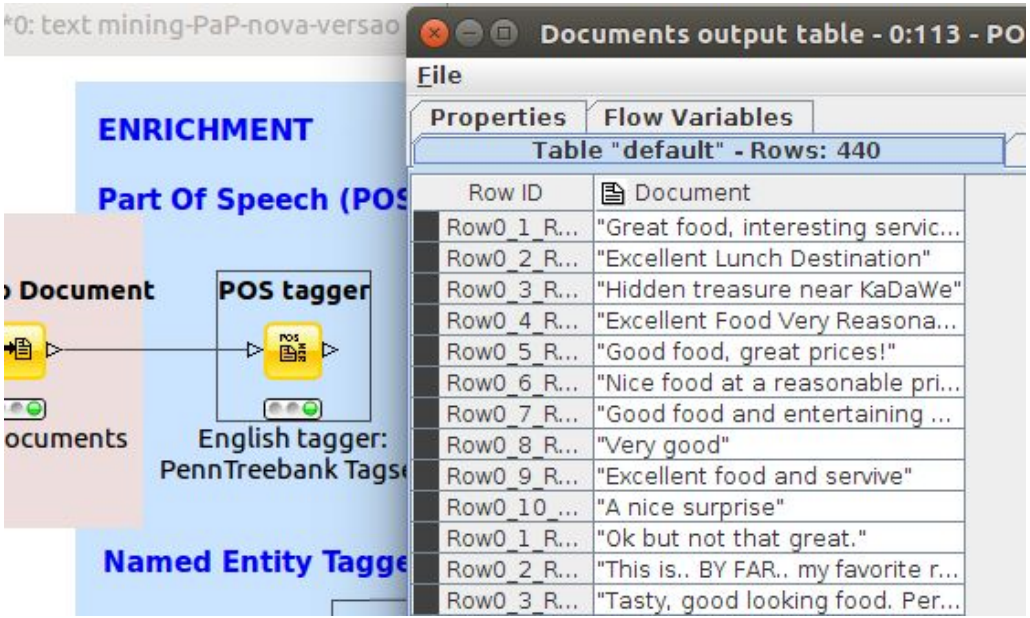
- Utiliza uma gramática para a língua inglesa e através de modelos probabilísticos procura detectar classes de palavras: adjetivos, verbos, sujeitos...

Enriquecimento (Inglês)

❖ Part Of Speech (POS-Tagging)

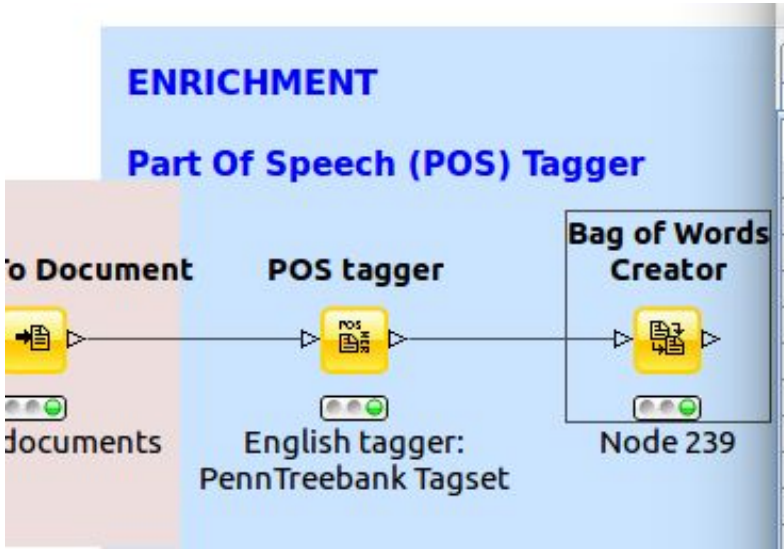
O Retorna apenas a coluna Document.

Como podemos extrair e visualizar termos de um documento?



Enriquecimento (Inglês)

❖ Visualizando termos



File

Properties | Flow Variables

Table "default" - Rows: 21290 | Spec - Colu

Row ID	Term	Document
Row1	Great[JJ(POS)]	"Great food, interesting
Row2	food[NN(POS)]	"Great food, interesting
Row3	,[SYM(POS)]	"Great food, interesting
Row4	interesting[JJ(POS)]	"Great food, interesting
Row5	service[NN(POS)]	"Great food, interesting
Row6	this[DT(POS)]	"Great food, interesting
Row7	restaurant[NN(POS)]	"Great food, interesting
Row8	is[VBZ(POS)]	"Great food, interesting
Row9	seemingly[RB(POS)]	"Great food, interesting
Row10	known[JJ(POS)]	"Great food, interesting
Row11	by[IN(POS)]	"Great food, interesting

Observe que agora temos os colchetes preenchidos:

- **JJ,NN,VBZ... :** A classe gramatical detectada
- **POS:** O algoritmo que fez a deteccção

Enriquecimento (Inglês)

- **Part Of Speech**

Referência - Pag 6

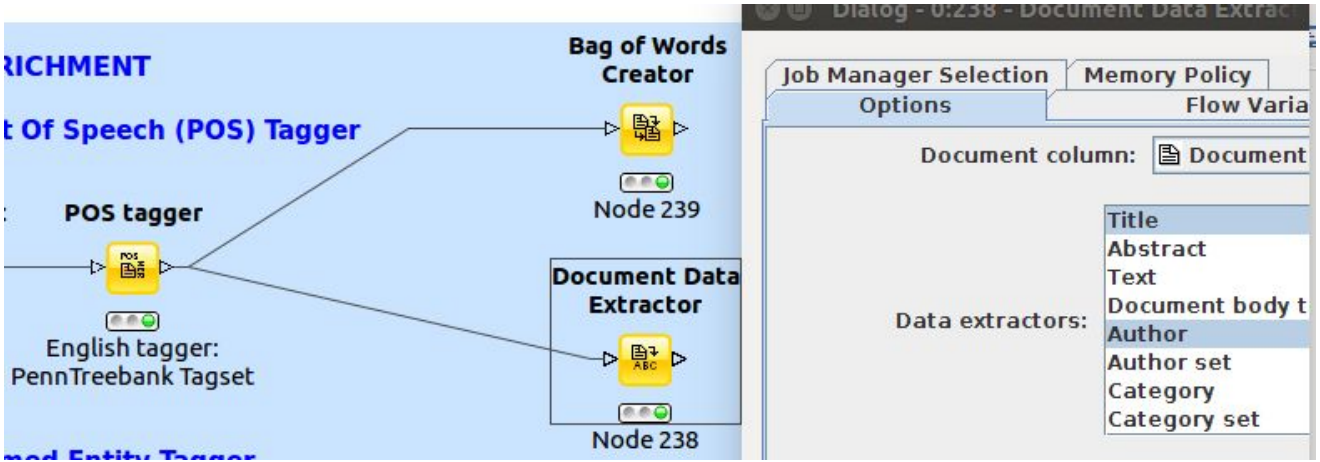
3 List of tags with corresponding part of speech

This section contains a list of tags in alphabetical order and the parts of speech corresponding to them.

1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NP	Proper noun, singular
15.	NPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PP	Personal pronoun
19.	PP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Enriquecimento (Inglês)

❖ Metadados dos Documentos



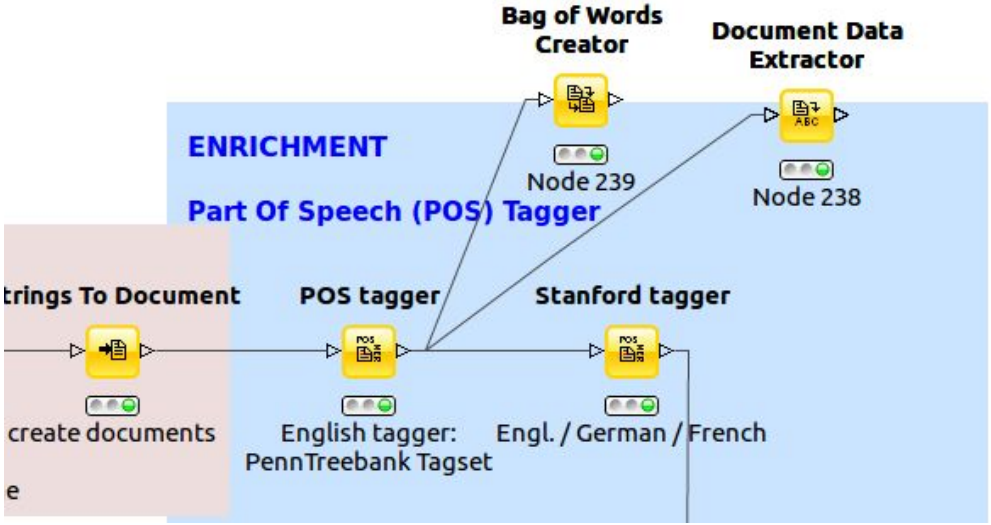
Para gerar uma tabela com os metadados dos documentos basta utilizar o Nó **Document Data Extractor** e selecionar as colunas de interesse

Enriquecimento (Inglês)

Para otimizar nosso fluxo vamos processar apenas a coluna Document e deixar a extração de termos/metadados para a fase de pré-processamento e análise

Enriquecimento (Inglês)

❖ Part Of Speech (POS-Tagging)



➢ Podemos utilizar nodes em sequência

É possível configurar modelos para Inglês, Alemão e Francês

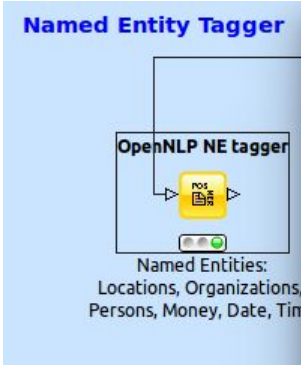
➢ Nó **Stanford tagger**: Assim como o **POS tagger** procura detectar classes de palavras.

➢ Referência: http://nlp.stanford.edu/software/tagger.shtml

Detecção de Entidades - Inglês

❖ Detecção de Entidades

O node **OpenNLP NE Tagger** tenta reconhecer algumas categorias de entidades no texto.



Qual a entrada de dados adequada desse node?

- Entidades reconhecidas: Persons, Locations, Organizations, Money, Date, and Time. Configure para Person.
- Referência: OpenNlp natural language processing toolkit

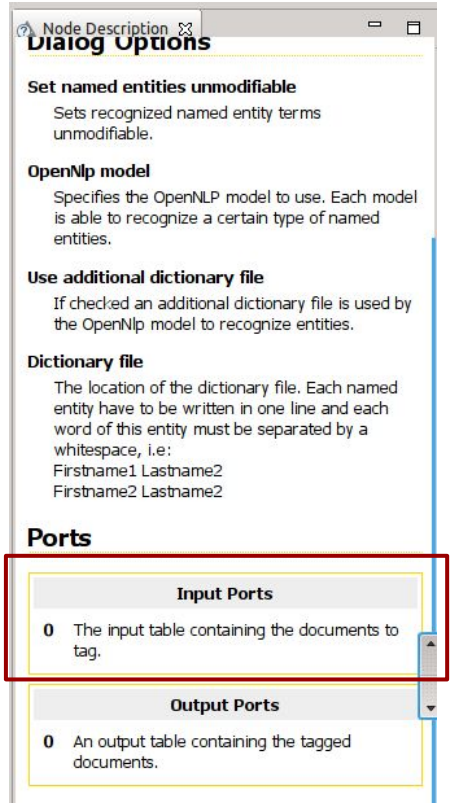
Detecção de Entidades - Inglês

❖ Na dúvida verifique a referência

A entrada do **OpenNLP**, assim como dos **POS taggers**, são **documentos**.

Lembre-se da aba de referência dos nodes e de pensar no sentido de cada ação durante a montagem do fluxo

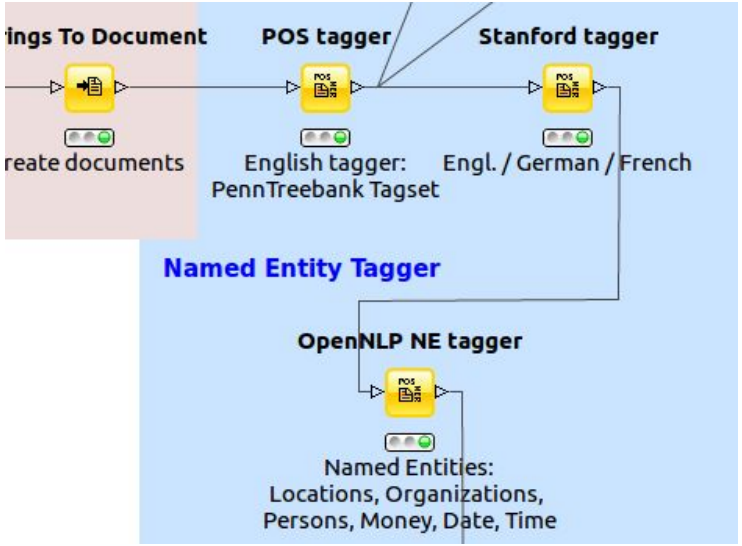
O enriquecimento de termos deve ser incremental (um node após o outro) ou paralelo (cada node em caminhos separados) ?



Detecção de Entidades - Inglês

❖ Detecção de Entidades

O node **OpenNLP NE Tagger** tenta reconhecer algumas categorias de entidades no texto.

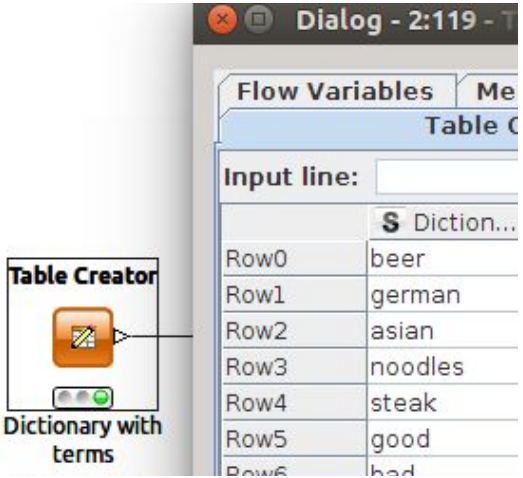


- Entidades reconhecidas: Persons, Locations, Organizations, Money, Date, and Time. Configure para Person.
- Referência: OpenNlp natural language processing toolkit

Dicionários de Termos

❖ Dicionários Próprios de Entidades

Podemos criar (ou ler) uma tabela com conjuntos de termos de interesse a serem destacados no texto

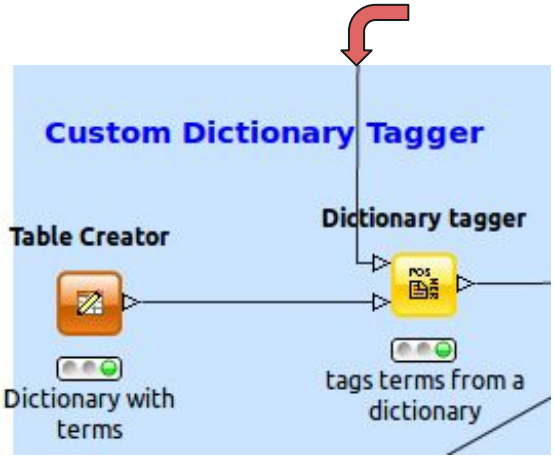


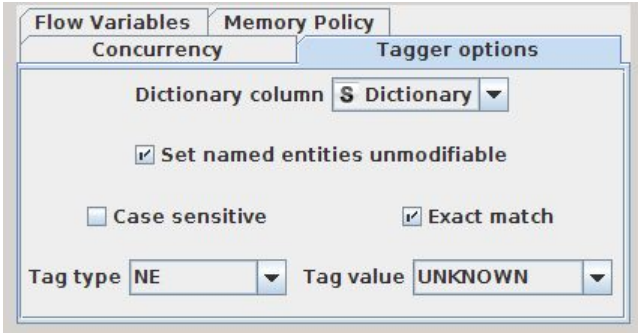
Dicionários de Termos

❖ Dicionários Próprios de Entidades

O **Dictionary Tagger** atribui uma tag sempre que encontrar alguma palavra da tabela no texto dos documentos.

Portanto, além da tabela de termos, qual é a primeira entrada do Dictionary Tagger?



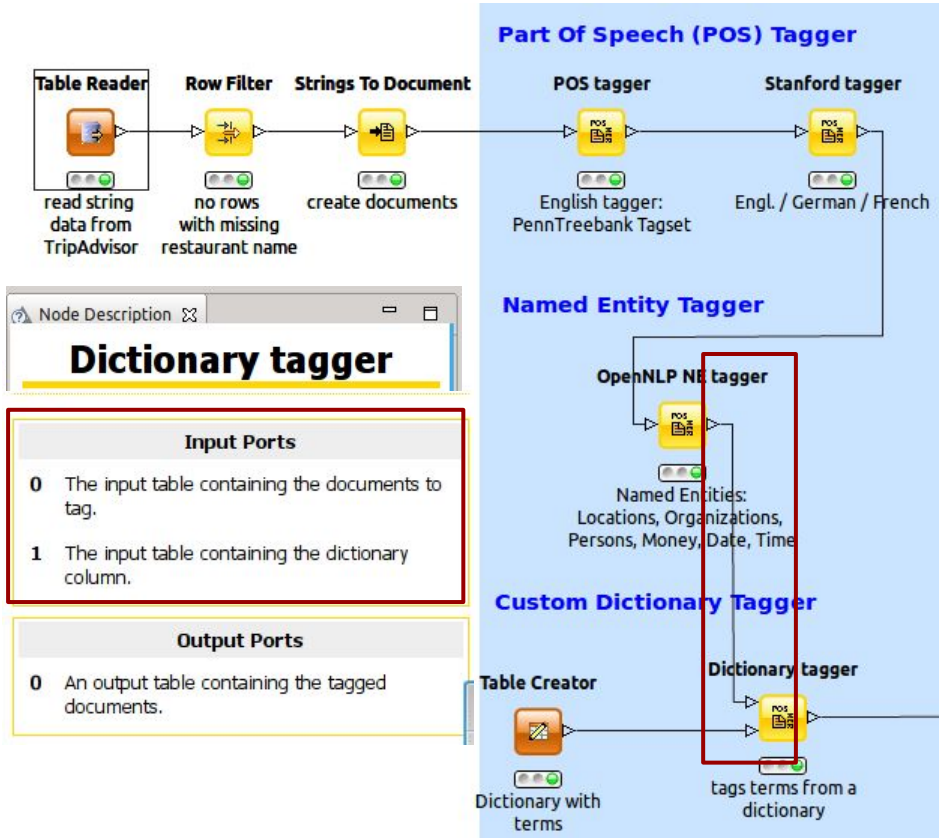


Dicionários Próprios de Entidades

❖ Dicionários de Entidades

A primeira entrada do Dictionary Tagger é o documento que possui o texto

A segunda é o conjunto de termos que serão procurados no texto



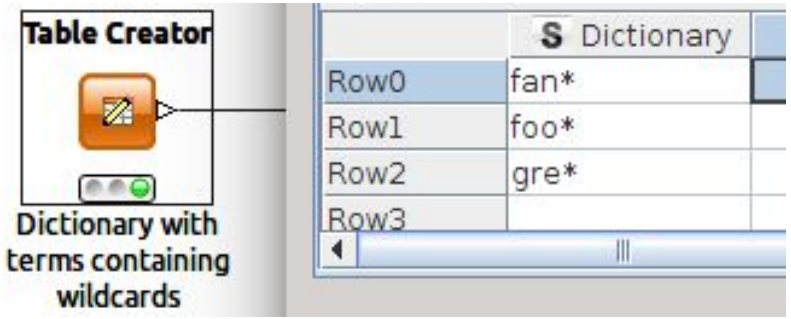
Dicionários Próprios de Entidades

❖ Dicionários de Wildcards

Wildcards são tipos de regras onde:

O “*” indica qualquer conjunto de caracteres que termine a palavra.

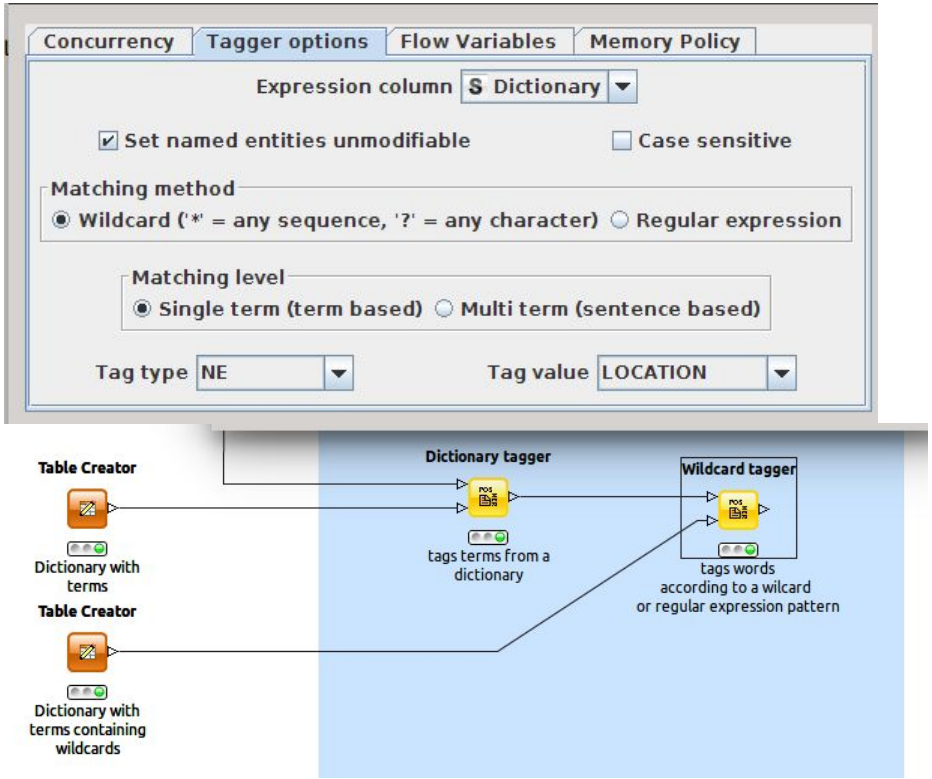
A “?” indica apenas um outro caractere.



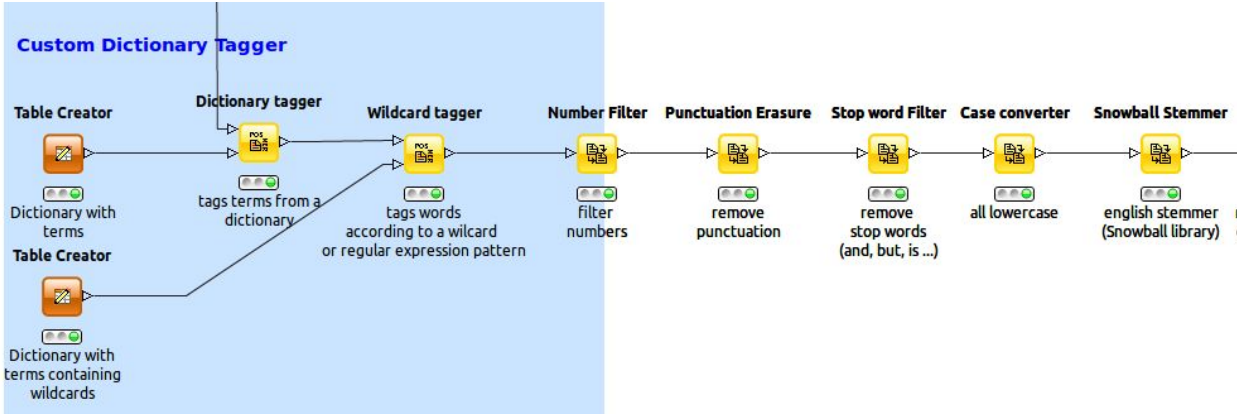
Ex: “gre*” = “great, greatest, greater…
Lembre-se que é **qualquer** caractere, ou seja, gre9, gremlins, etc.. também são válidos

Wildcards

O **Wildcard Tagger** faz a tarefa de procurar os termos no texto conforme as regras da tabela de wildcards previamente criada e marcar essas palavras com tags que permitem filtragens futuras.



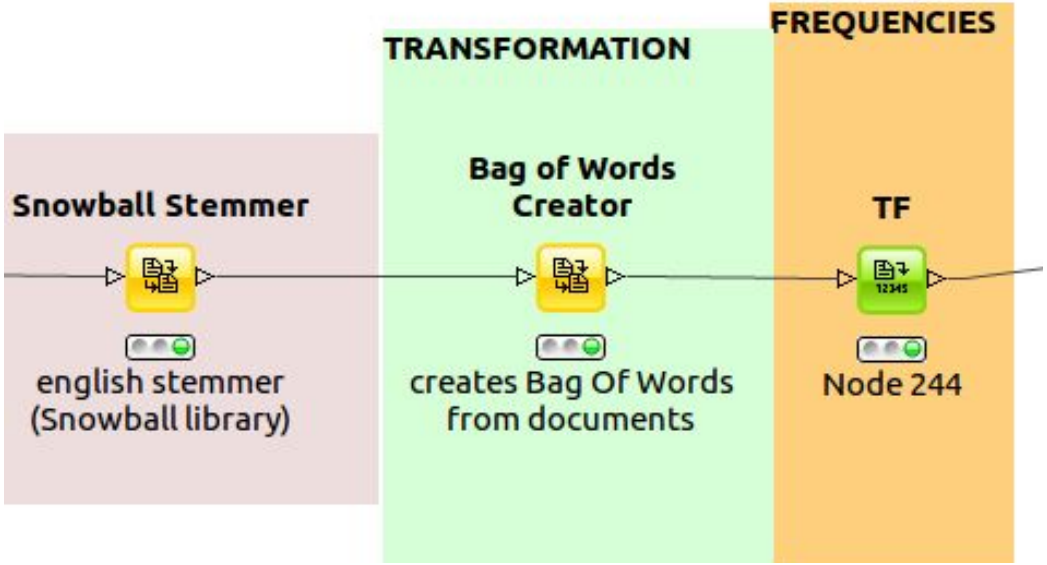
Pré-processamento



Note que também é possível pré-processar o texto **antes** de criar a tabela de **Bag of Words**

Por que identificamos entidades e classes de palavras antes do pré-processamento?

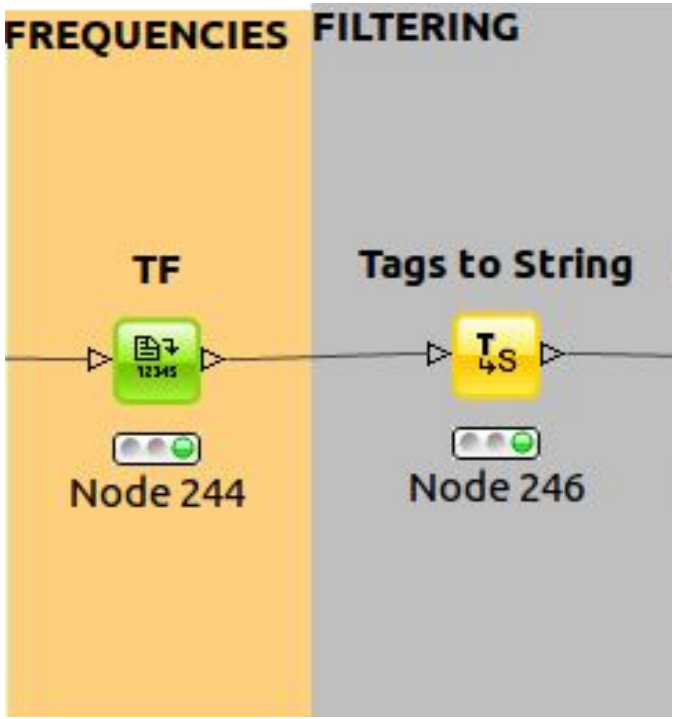
Transformação e Frequências



Qual a vantagem do bag of words no fim?

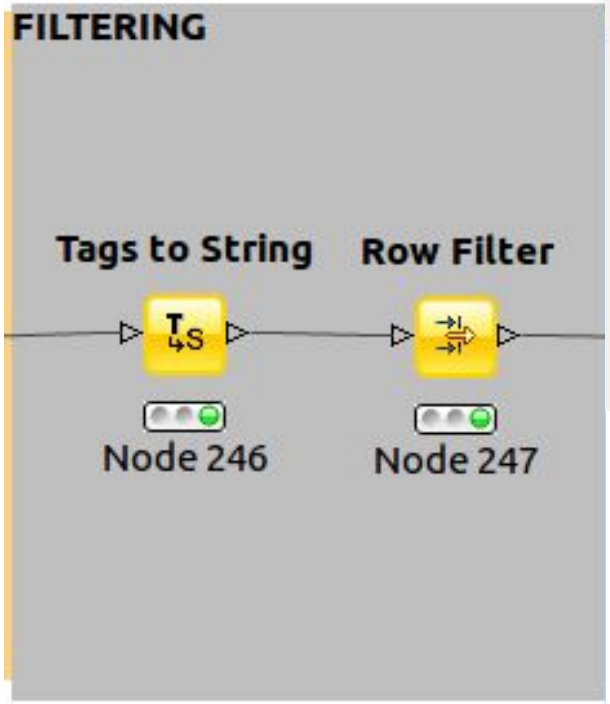
Vamos gerar a contagem de termos com **Bag of Words** + **TF**

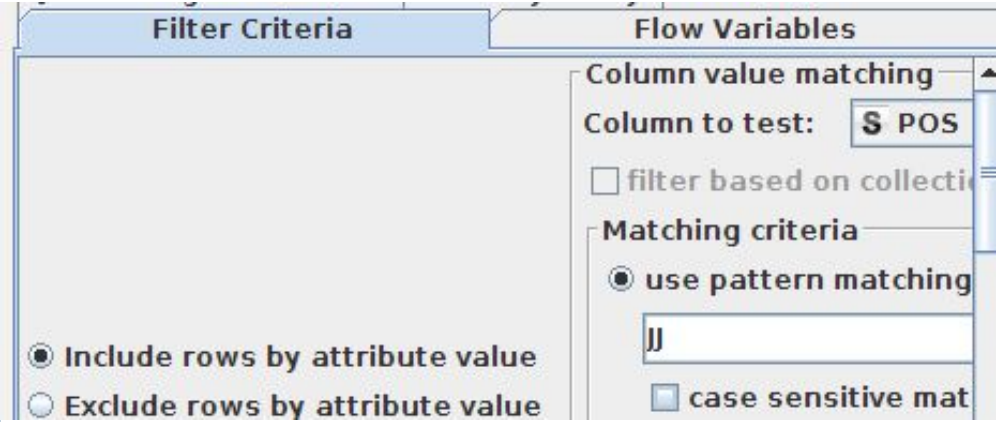
Visualização



É possível extrair as **Tags** e transformá-las **em strings** para aplicação de filtros ou manipulações futuras.

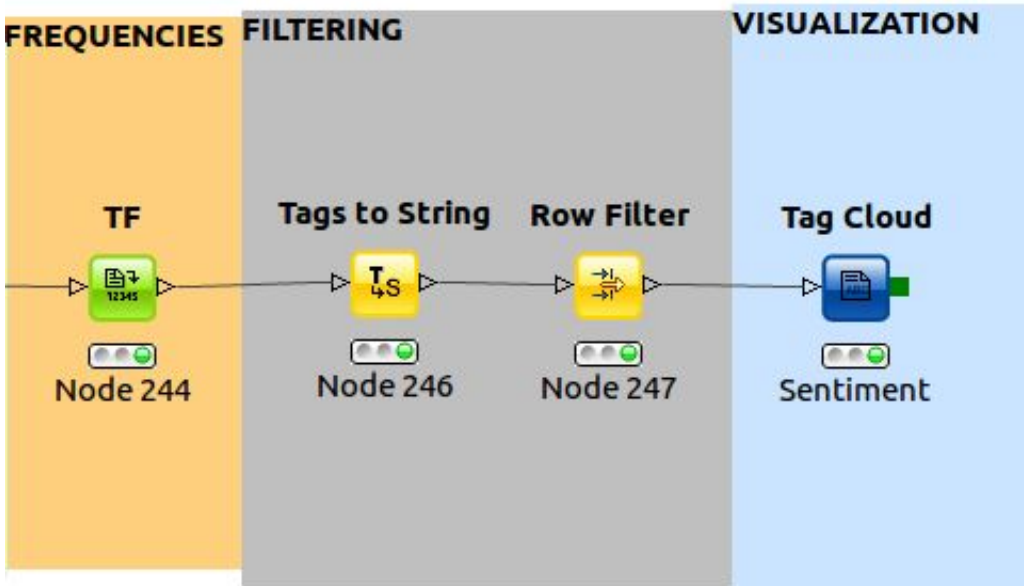
Visualização





Filtrando a **Tag** JJ

Visualização



Por fim visualizamos os termos utilizando uma **Tag Cloud**

KNIME - Pré-processamento

Extração de Texto por Regras

- Expressões Regulares ou Regex (notações)

metacaractere	conhecido como	significado
`.`	curinga	qualquer caractere, exceto a quebra de linha `\n` (ver *flag_dotall*)
`[...]`	conjunto	qualquer caractere incluido no conjunto
`[^...]`	conjunto negado	qualquer caractere não incluido no conjunto
`\d`	dígito	o mesmo que `[0-9]`
`\D`	não-digíto	o mesmo que `[^0-9]`
`\s`	branco	espaço, quebra de linha, tabs etc.; o mesmo que `[ \t\n\r\f\v]`
`\S`	não-branco	o mesmo que `[^ \t\n\r\f\v]`
`\w`	alfanumérico	o mesmo que `[a-zA-Z0-9_]` (mas pode incluir caracteres Unicode; ver *flag_unicode*)
`\W`	não-alfanumérico	o complemento de `\w`
`\`	escape	anula o significado especial do metacaractere seguinte; por exemplo, `\.` representa apenas um ponto, e não o curinga

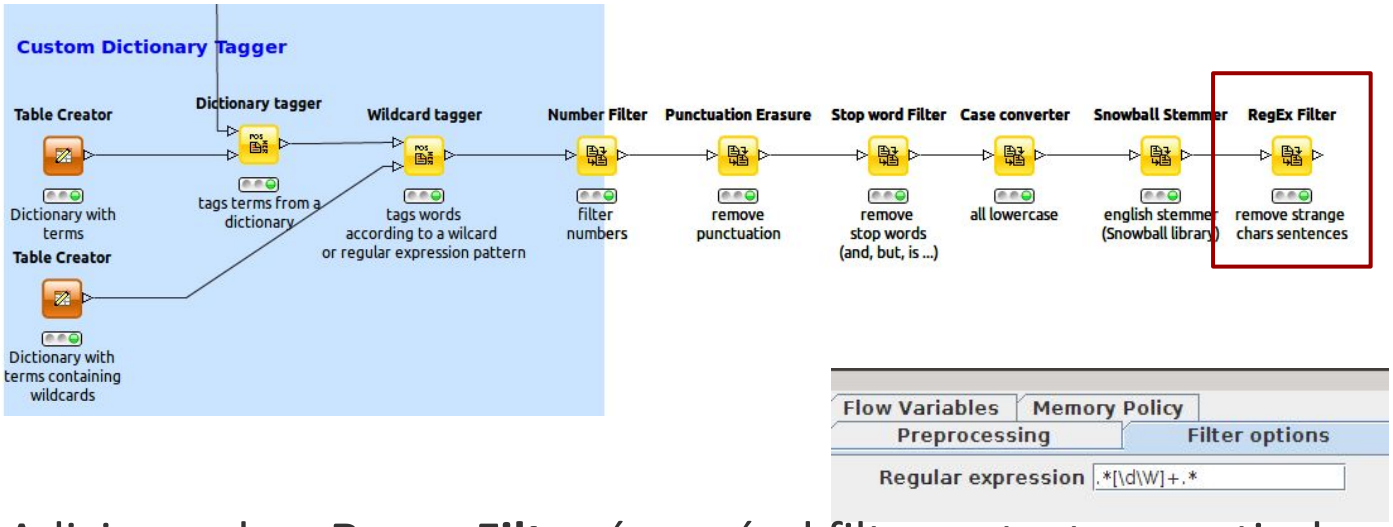
Semelhante aos wildcards as expressões regulares permitem a criação de diversos tipos de regras para tratar o texto.

Extração de Texto por Regras

❖ Exemplos simples (referência)

- ➢ A RegEx **[c9]** reconhece o padrão **c** e o padrão **9**
- ➢ A RegEx **[A-Z]** reconhece **todas** as **letras maiúsculas**
- ➢ A RegEx **[A-Z0-9]** reconhece todas as **letras maiúsculas** e **números**
- ➢ A classe **\d** é equivalente à classe **[0-9]**

❖ Pré-processamento

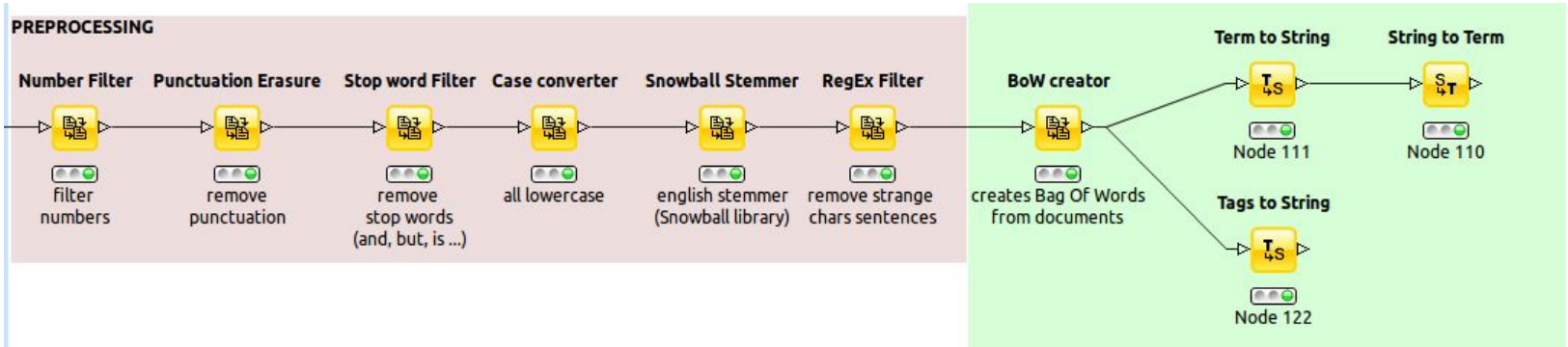


Adicionando o **Regex Filter** é possível filtrar o texto a partir de regras. Nesse caso eliminamos strings que possuem dígitos (\d) e caracteres que não são letras (\W)

KNIME - Transformação

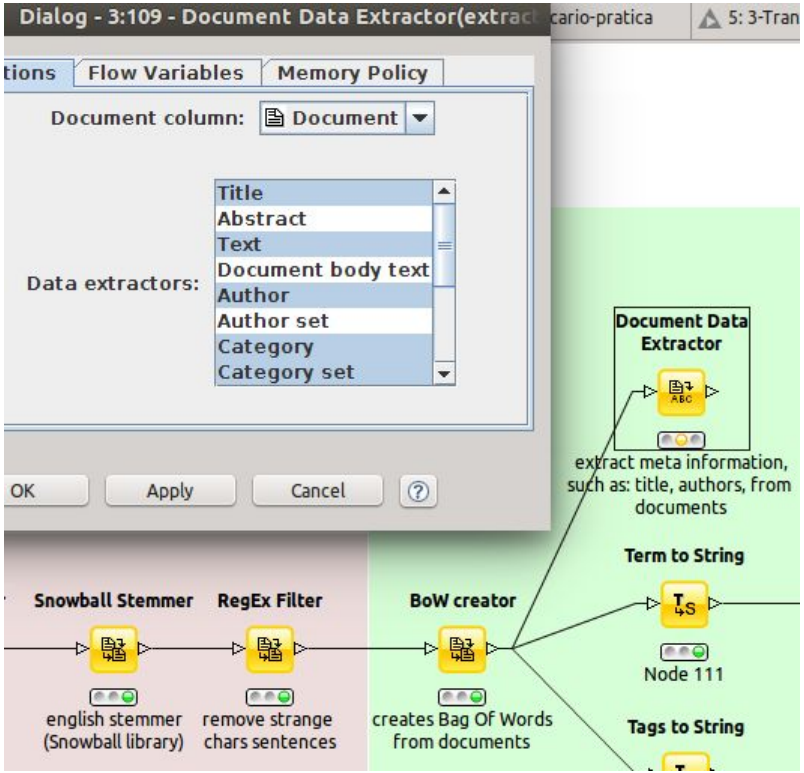
Transformação

Além da criação do conjunto de palavras (**Bag Of Words**), sempre que necessário será possível transformar termos em strings e vice versa. O mesmo com tags dos termos. Os nodes são *Term to String, String to Term* e *Tags to String*. Com o *Tags to String* é possível definir o tipo de tag a ser transformada em string.



Transformação

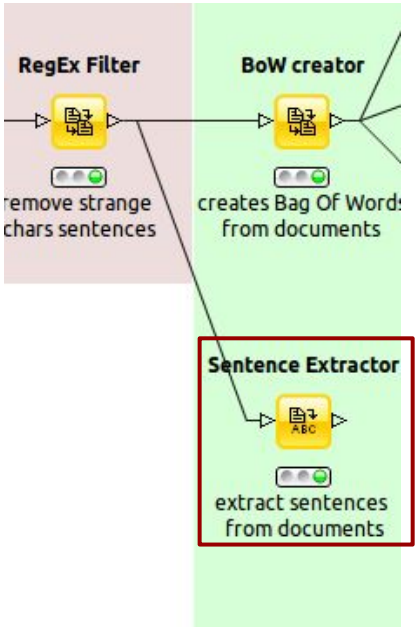
De posse de uma coluna contendo os documentos, sempre é possível resgatar outros atributos com o node *Document Data Extractor*



Transformação

Além de palavras...

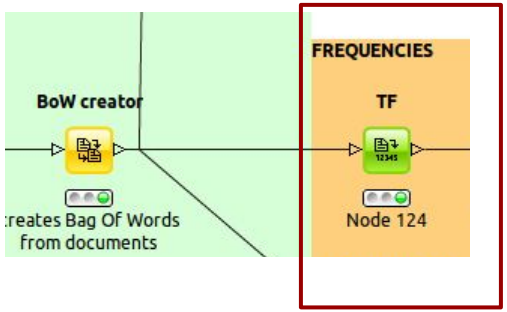
O node **Sentence Extractor** procura extrair e identificar frases no texto.



KNIME - Frequência de Termos

Frequência de Termos

Além de contagem de palavras com **TF** e grau de raridade com **IDF**. Podemos identificar sequências importantes de palavras.

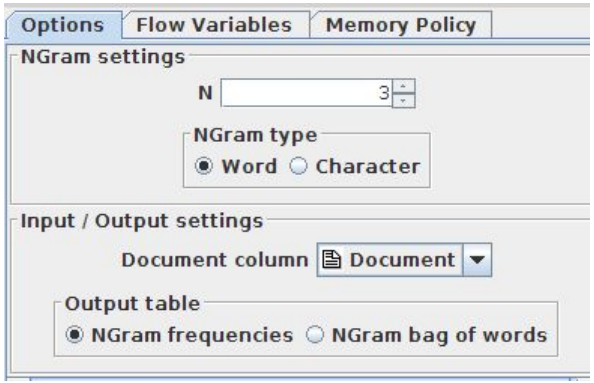


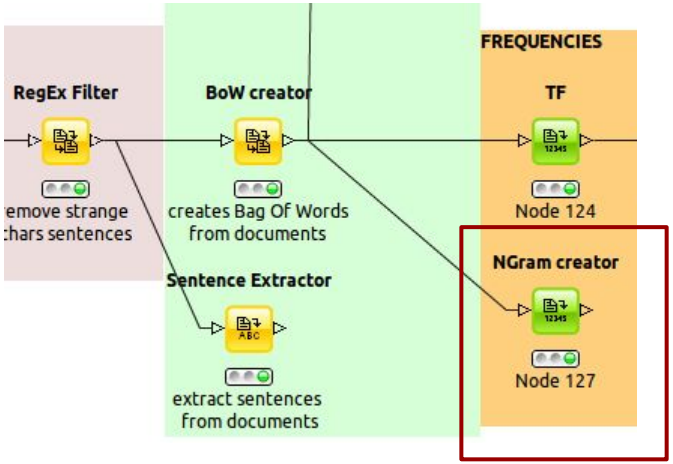
Para isso extraímos **n-gramas** no texto.

Um *n-grama* é uma sequência de *n* letras ou palavras. Para **n=2** temos conjuntos de palavras duas a duas ou **bigramas**. Para **n=3** temos conjuntos de palavras três a três ou **trigramas**, e assim por diante.

Frequência de Termos

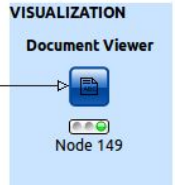
É possível extrair Ngramas com o **NGram creator** e determinar o tamanho de N em *duas a duas, três a três….*

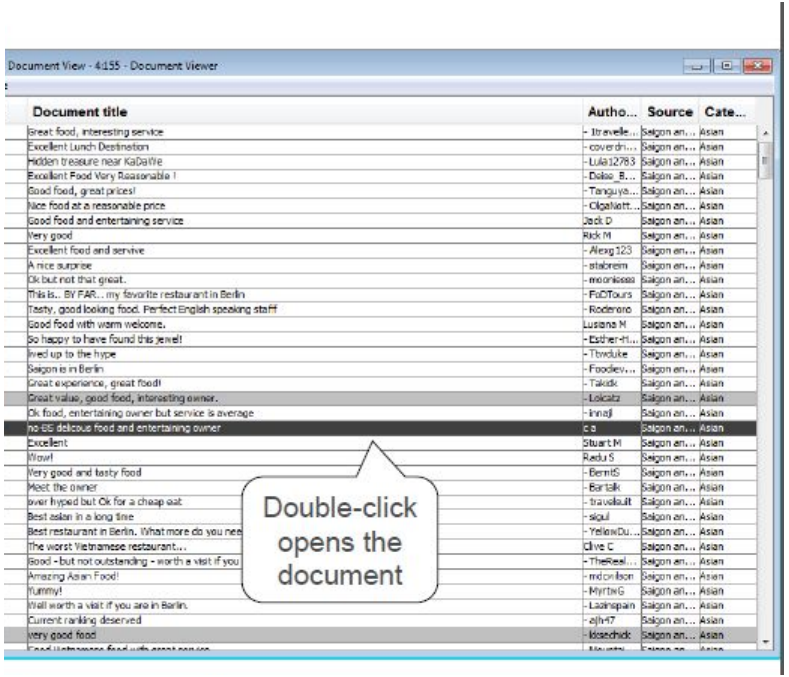




KNIME - Visualização

Visualizando Documentos





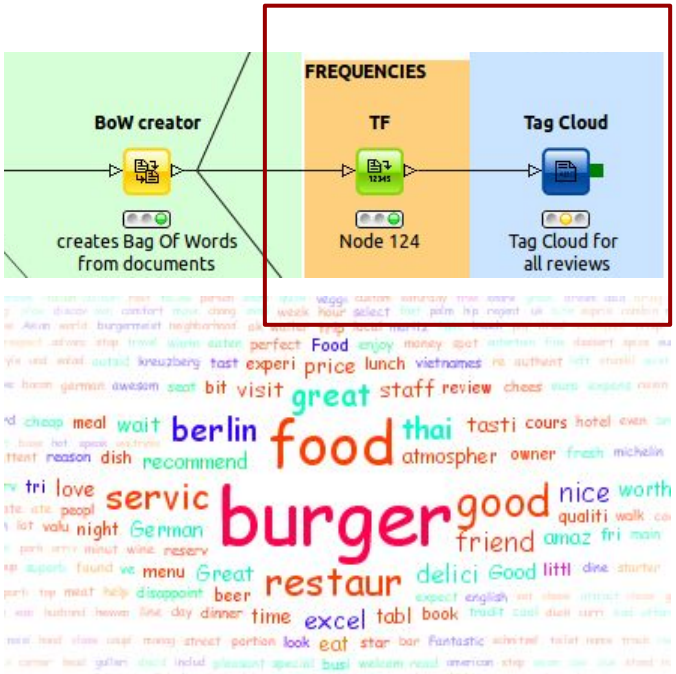
Para visualizar documentos basta contectar qualquer saida que possua uma coluna com documentos

Atenção: utilize colunas de valores únicos de documentos, do contrário a lista para visualização terá documentos repetidos

Nuvem de Palavras

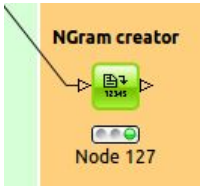
Com o node **Tag Cloud** fica muito(!!) mais intuitivo e fácil analisar as palavras mais comuns. É uma das técnicas simples e eficientes para visualização em Text Mining.

Basta conectar uma tabela que possua um atributo numérico para cada termo, como a contagem por TF, IDF, etc...



Nuvem de Palavras

Crie uma tagcloud com ngramas

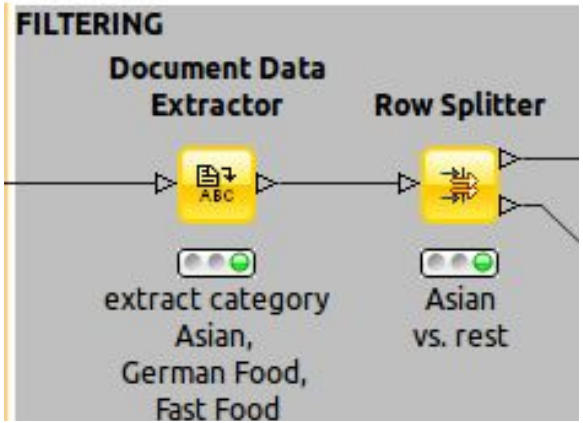


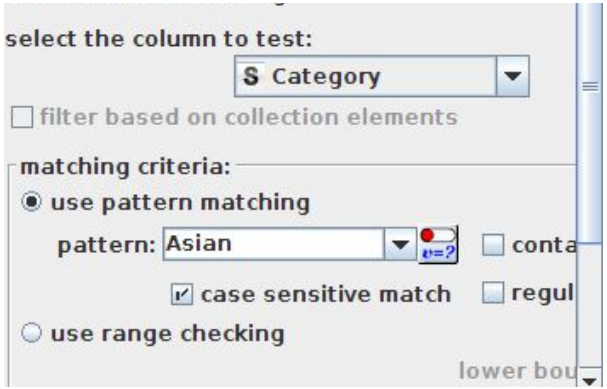
Nota: vai dar erro na tentativa de ligar diretamente ao *TagCloud.*
Por que?

Filtros

Extraia a categoria dos documentos.

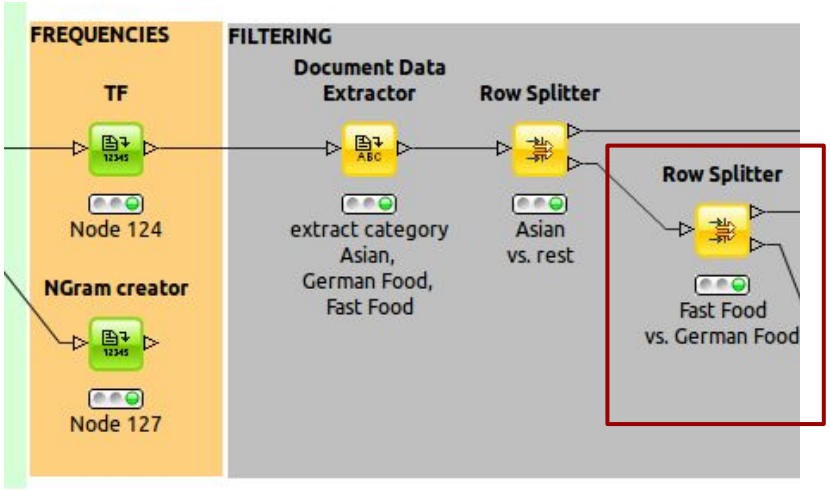
O **Row Splitter** separa os dados de acordo com o valor das linhas para o atributo. Nesse caso “Category”.





Separe os documentos em documentos da categoria “Asian” dos demais

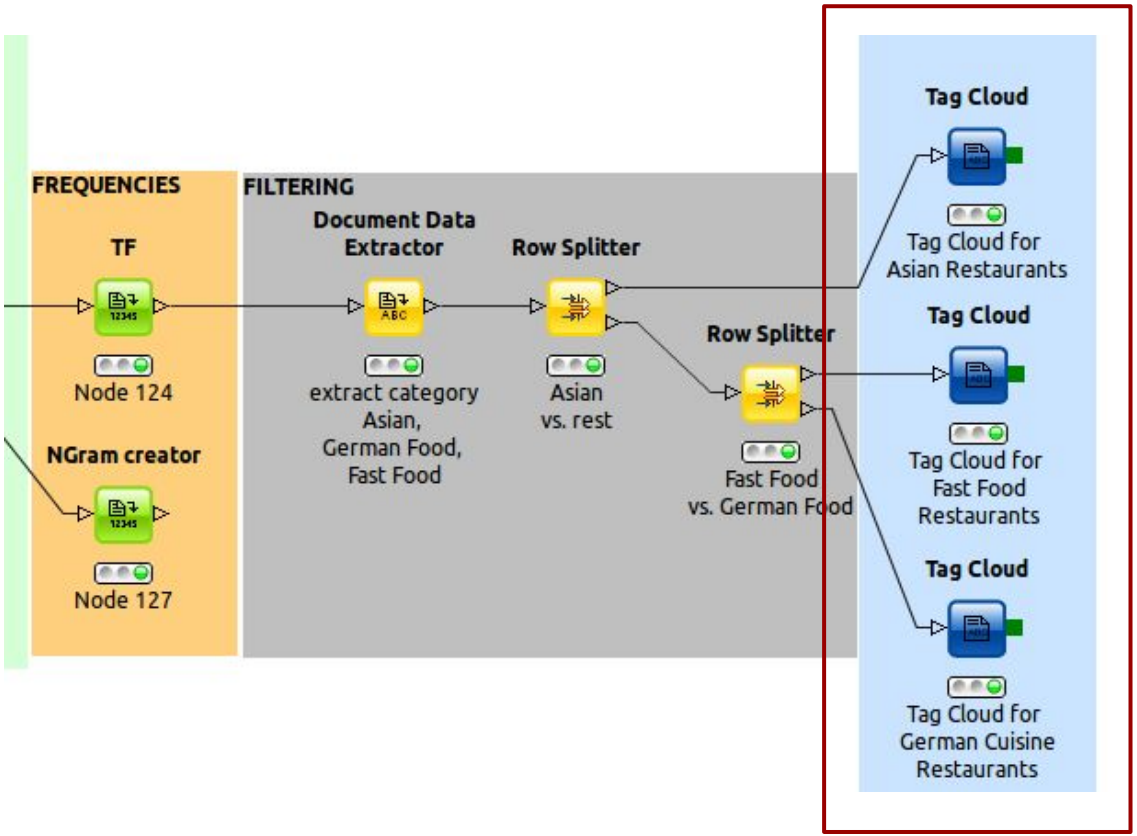
Filtros



Use o mesmo método para filtrar fastfood a partir do que não é “Asian”

Como são apenas 3 categorias, pós filtrar "fastfood” o que sobrar é da categoria restante “German”

Visualização

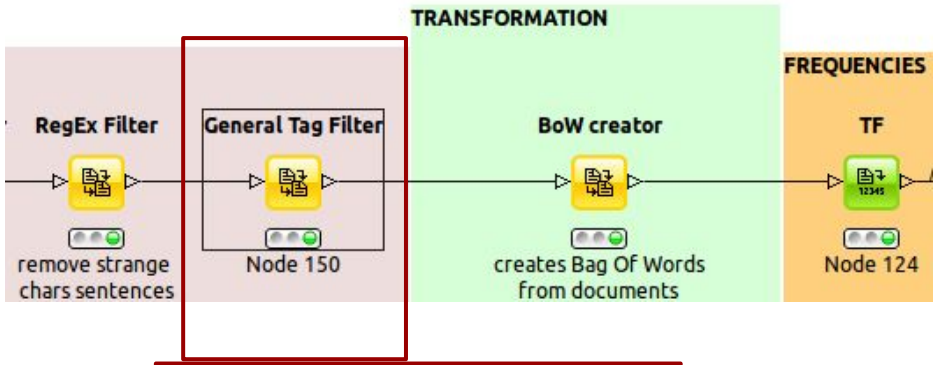


Crie visualizações de Tag Clouds para os documentos de cada categoria.

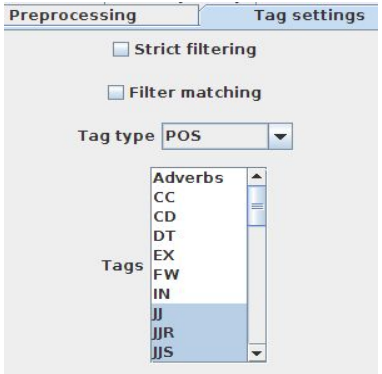
Filtros de Tags

Com o **Tag Filter** é possível filtrar as tags criadas na etapa de Enriquecimento

Normalmente isso é feito antes da contagem dos termos (ex: TF)



Utilize o **Tag Filter** ao invés do **General Tag Filter**



Filtrando apenas adjetivos

JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative

Referência - Pag 6

Filtros de Tags

- Filtrando e Visualizando apenas adjetivos

KNIME - Sentimento

Sentimento

❖ Análise de Sentimento

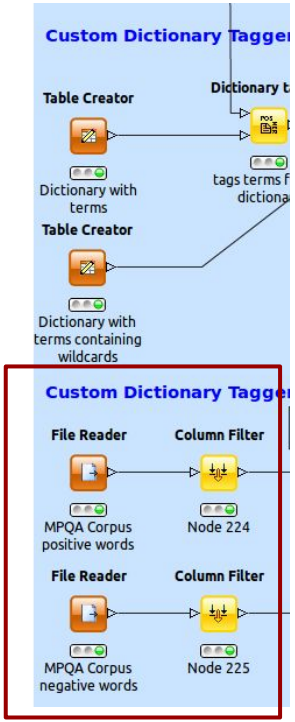
- ➢ Na literatura existem **dicionários** de termos com a **polaridade** de palavras
- ➢ Podemos usar um dicionário para identificar esses termos nos documentos coletados
- ➢ Além disso podemos tentar avaliar a polaridade de um documento inteiro através de **operações** (até mesmo aritiméticas) sobre esses termos

Sentimento

❖ Análise de Sentimento

Com o **File Reader** podemos ler arquivos de dados. Nesse exemplo é carregamos um arquivo *.csv* com palavras previamente categorizadas com polaridade positiva e negativa

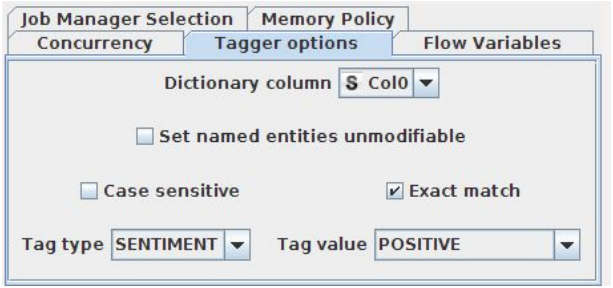
Com o **Column Filter** filtramos a coluna de palavras em Inglês (col0). A segunda contém palavras em português (col1).



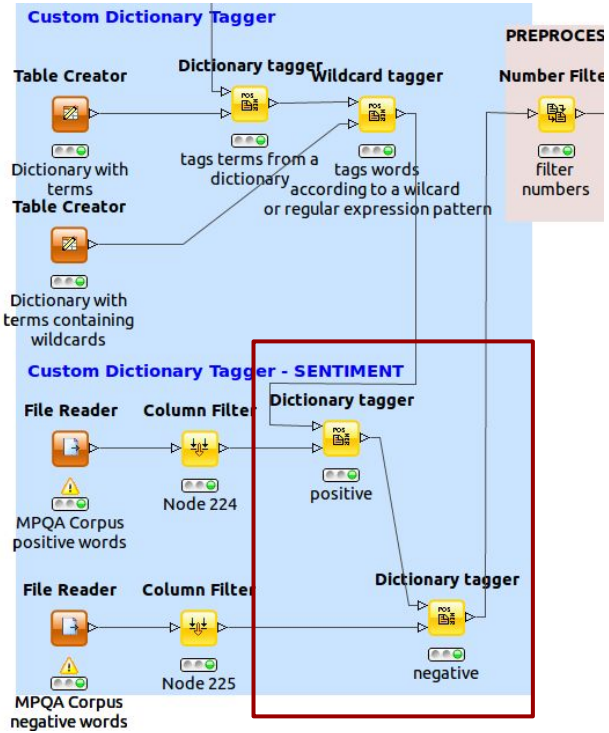
Sentimento

❖ Análise de Sentimento

Usando a mesma estratégia de dicionários, utilizando o **Dictionary Tagger** podemos mapear termos positivos e negativos.



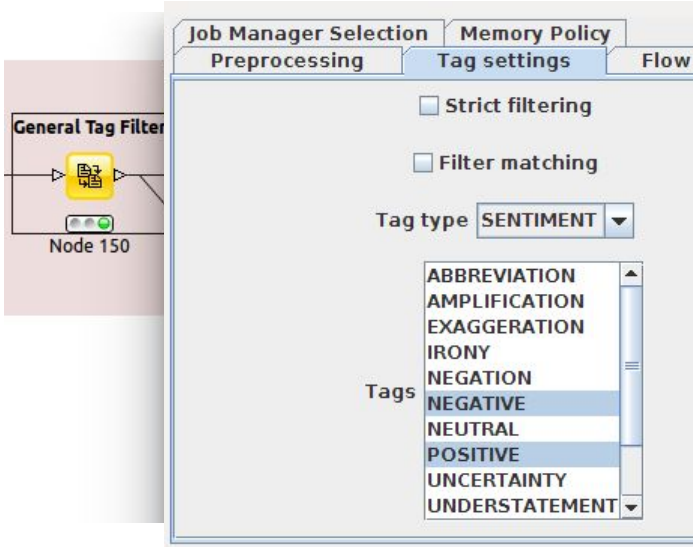
Criando tags para palavras positivas



Sentimento

❖ Análise de Sentimento

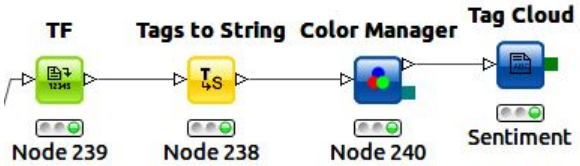
Ao fim do **processamento** do termos filtramos as tags POSITIVE e NEGATIVE com o **General Tag Filter.**

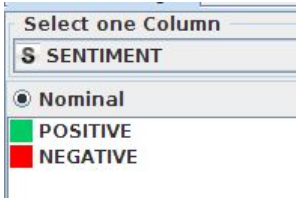


Sentimento

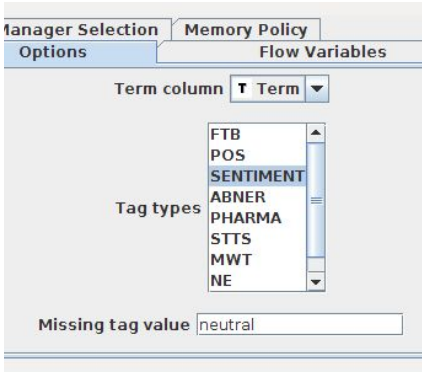
❖ Análise de Sentimento

Após a **contagem** de termos, para melhor visualização da TagCloud, podemos definir cores com o **Color Manager** para cada tag de sentimento.





Nesse caso é necessário transformar as tags em strings com o **Tags to String.**



Sentimento

❖ Análise de Sentimento

É possivel usar tagcloud para refinar classe negativa e positiva de termos ao usar em próximas análises

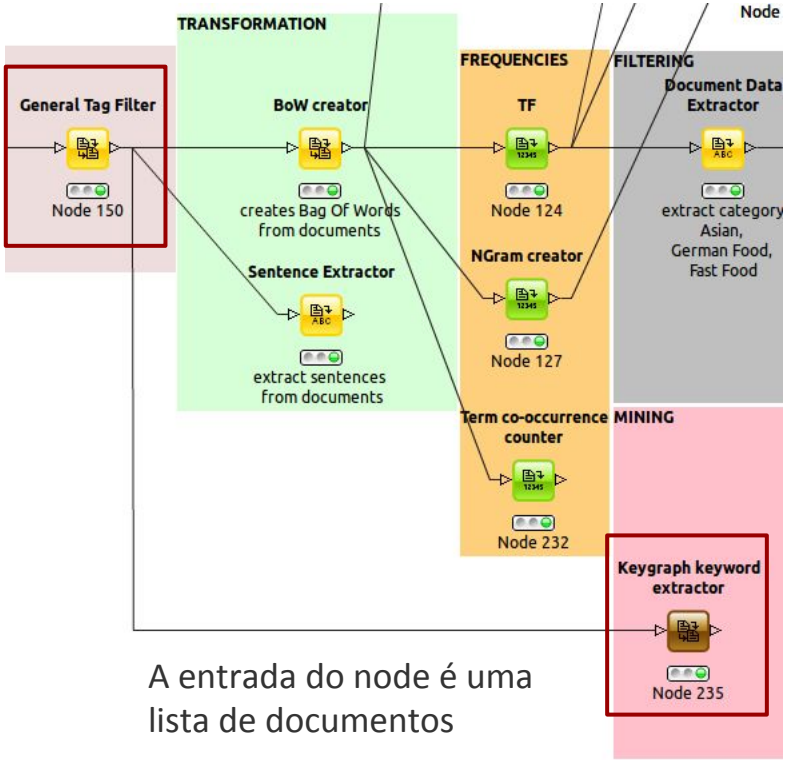
KNIME - Palavras-Chave

Palavras-chave

❖ Palavras-chave

Palavras chave é uma alternativa as contagens simples de frequência.

O **Keygraph Keyword Extractor** utiliza uma estratégia baseada em grafos (próxima aula) para mapear coocorrência entre termos nos documentos e descobrir possíveis palavras chave. (Vide aba de referência)



A entrada do node é uma lista de documentos

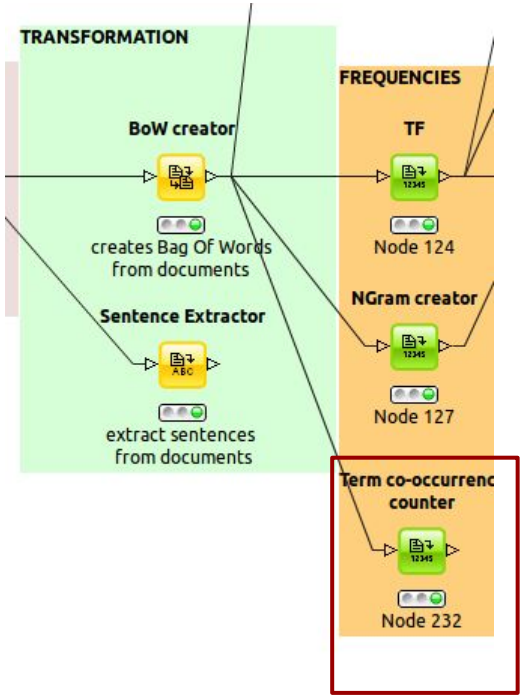
Palavras-chave

Crie uma visualização para palavras chave

KNIME - Coocorrência de Palavras

Coocorrência de Palavras

O node **Term co-ocurrence counter** é similar a contagem de **NGramas**, exceto que este não exige que as palavras apareçam uma após a outra no texto original. Basta aparecerem em conjunto dentro de diferentes níveis de coocorrência:



Coocorrência de Palavras

❖ Coocorrência de Palavras (Tarefa a ser entregue)

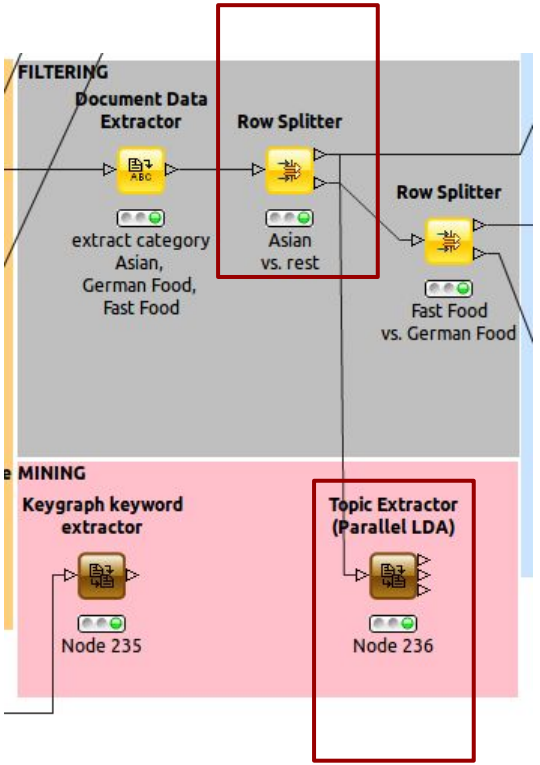
- **Crie uma visualização para coocorrência de palavras.**
  - Pensem em como trabalhar as colunas de termos para que seja possível gerar uma visualização de coocorrência de Palavras.

KNIME - Tópicos

Tópicos

❖ Mineração de Tópicos

O **Topic Extractor** tenta identificar palavras que possuem um contexto em comum e poderiam formar tópicos dentro de um documento.

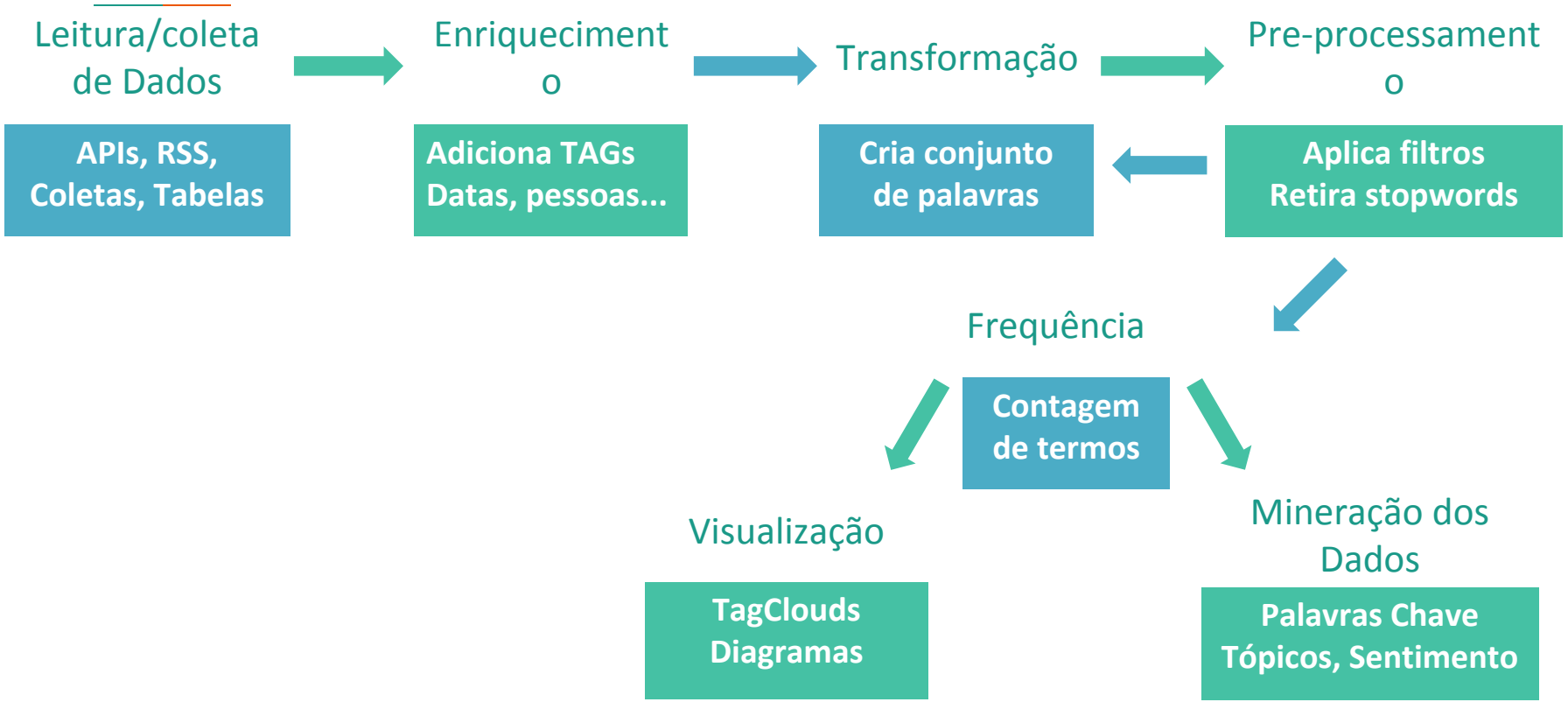


Esse processamento é muito caro do ponto de vista computacional, portanto procure analisar um conjunto menor de documentos previamente filtrados e tratados.

Ex:

- Documentos em uma categoria específica;
- Top N documentos dos termos mais frequentes...

Mineração de Textos



Testes com a Língua Portuguesa
Inspiração para Projeto Final

Fluxo para Prática - Enriquecimento

- Enriquecer palavras com tags semânticas
- Detectar entidades
- Configurar dicionários de entidades
- Análise de sentimento

Obs: Input e dicionários estão disponíveis no fluxo: *Boticario-Enriquecimento*

POS tagging - Português

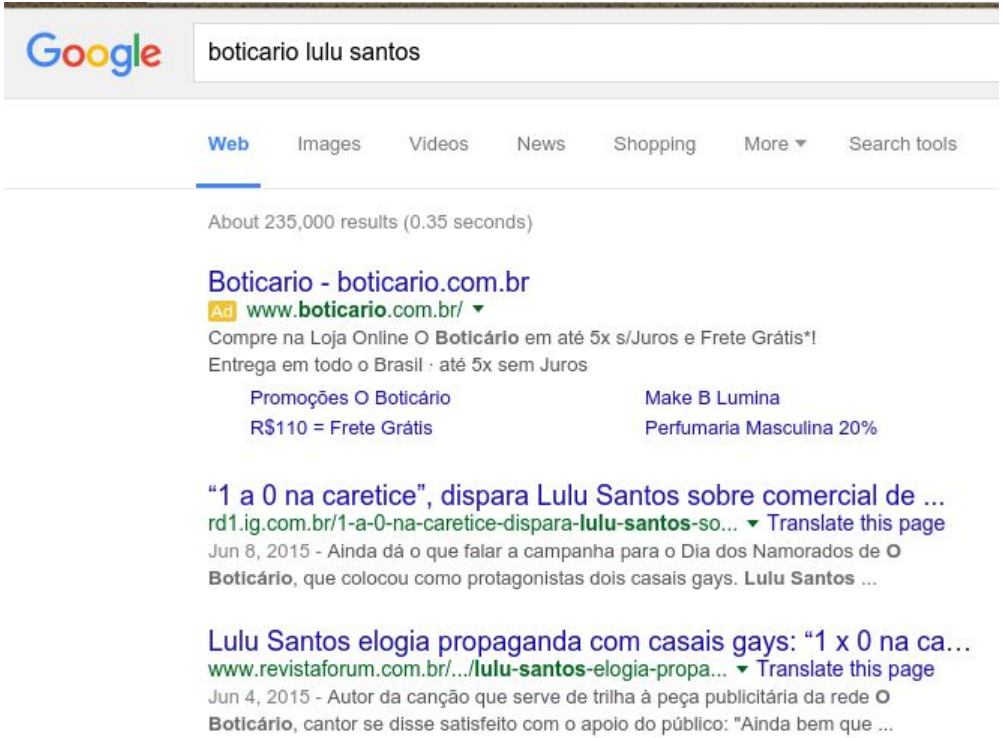
- **Filtragem de Adjetivos e Verbos nos Tweets**

Detecção de Entidades - Português

- **Detecção de Entidades: Pessoas**

Detecção de Entidades - Português

- **Lulu Santos?**



Detecção de Entidades - Português

- **Amplificadores (usando dicionário externo)**

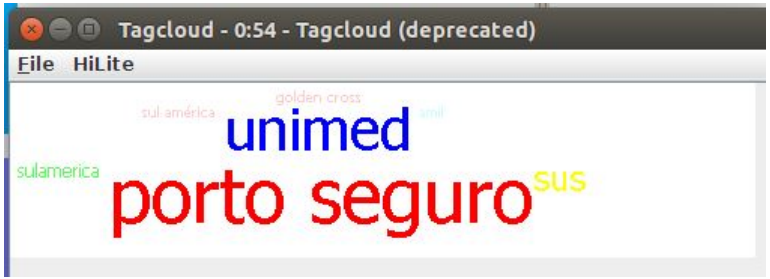
Sentimento - Português

❖ Sentimento

É possivel usar tagcloud para refinar classe negativa e positiva de termos ao usar em análises seguintes

Busca por Entidades

- Exemplo de procura pelas principais companhias de saúde mais citadas na rede



Análise de Sentimento

- Exemplo de sentimento das palavras chave encontradas no texto

A disciplina - RI

❖ Plano de Ensino

- **Unidade 01:** Conceitos de inteligência competitiva e coletiva, crowdsourcing e redes sociais. Recuperação da informação e Máquinas de busca. Desafios da Mineração na web e nas redes. Exemplos de Projetos da disciplina.

- **Unidade 02:** Algoritmos e soluções para problemas de busca e extração de informação da WWW. Ferramenta e prática de processamento textual e recuperação de informação.

- **Unidade 03:** Tipos de coleta, arquitetura e componentes de coletores Web. Ferramenta e prática de coleta de dados na Web.

A disciplina - RI

❖ Plano de Ensino

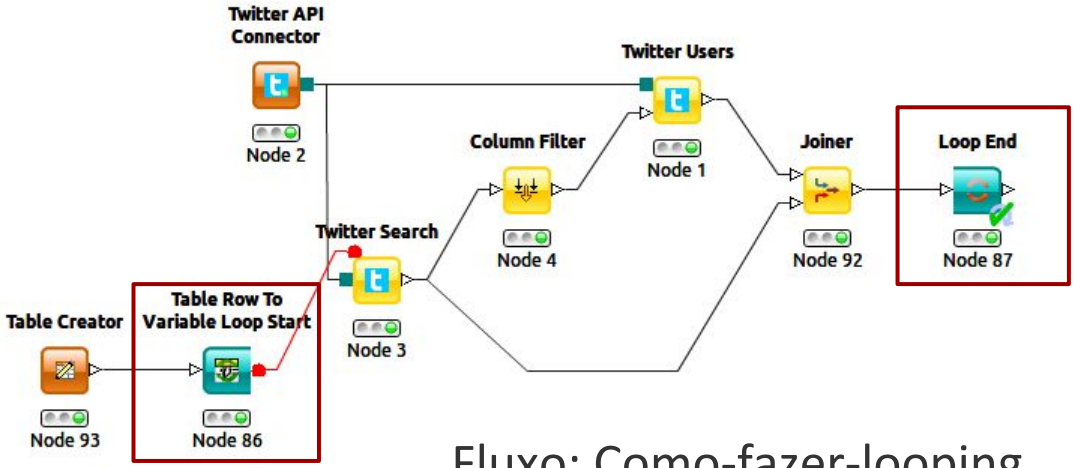
➢ **Unidade 04:** Aprofundando na mineração de texto e linguagem natural. Algoritmos e soluções para a análise da informação presente nas redes sociais online e em sites de conteúdo. Ferramenta e prática de mineração de texto.

Anexo - KNIME - Coleta de Vários termos (Looping)

Coleta de Vários termos (Looping)

O node *Table Row To Variable Loop Start* envia uma linha da tabela anterior a cada iteração

O node *Loop End* indica o fim do ciclo e armazena todos resultados gerados durante as iterações



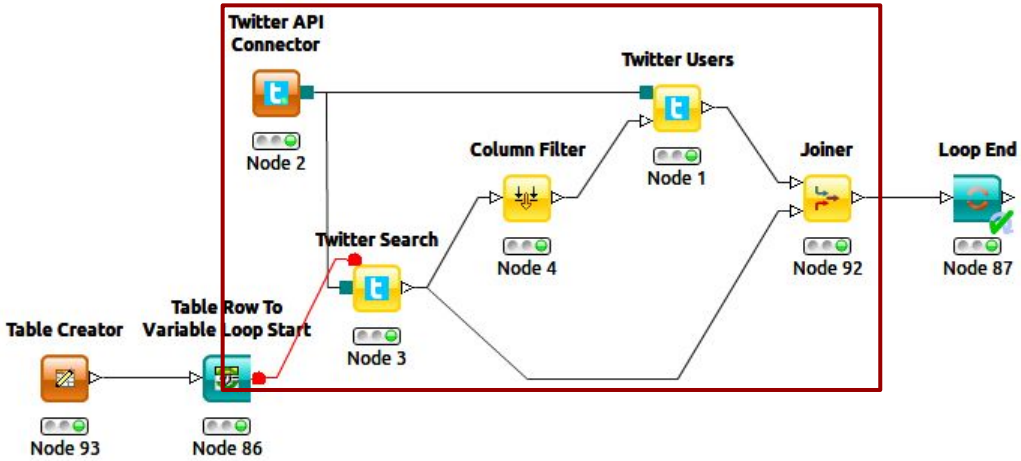
Fluxo: Como-fazer-looping

Coleta de Vários termos (Looping)

Todos os nodes entre o início e fim do looping são executados a cada passo.

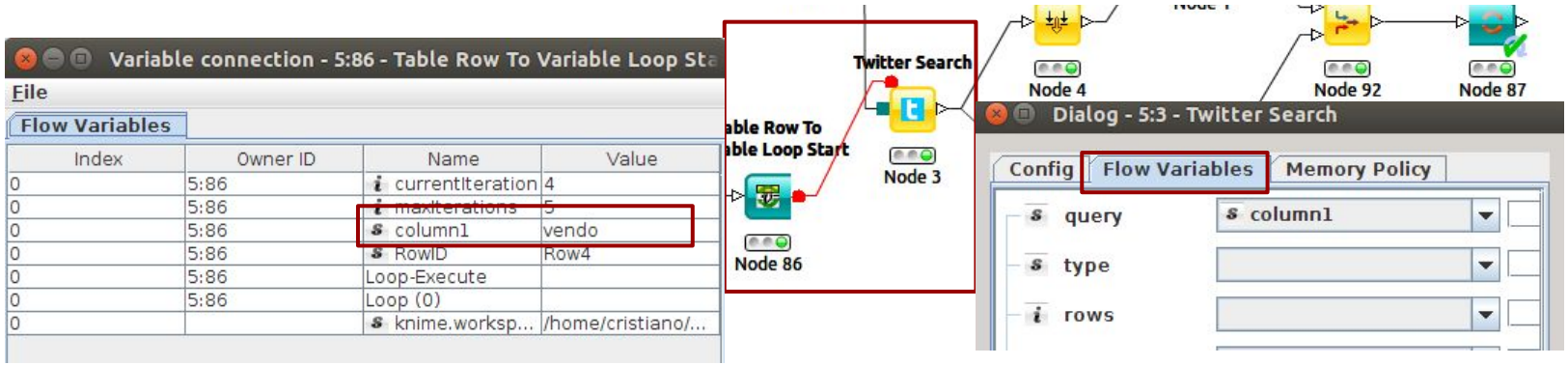
Ou seja, para cada linha da tabela de entrada.

Nesse caso, cada linha possui um termo e o ciclo faz a busca no twitter para cada termo



Coleta de Vários termos (Looping)

Em cada iteração o primeiro node envia a palavra para o node seguinte via “*flow variable*”




Thank
You
Obrigado
Mamana
GRAZIE
Juspaxar
KIITOS
MERCI
Vinaka
ASANTE
Kia Ora
DANK JE
SPASIBO
Arigato
MAAKE
Salamat
Matondo
NIRRINGRAZZJAK
Multumesc
Dankon
Maake
Ua Tsaug Rau Koj
Mochchakkeram
Terma Kasih
Raibh Maith Agat
Kiitos
Chokrane
Welalin
Nirringrazzjak
Matur Nuwun
Cam on ban
Merci
Grazie