IBM Watson

"more than a quiz show"

[redacted]

The Grand Challenge

Em 2007, a IBM Research encarou o grande desafio de construir um sistema que pudesse responder bem o suficiente, perguntas à máquina aberto para competir com competes no jogo "Jeopardy!".

[redacted]

DeepQA

A Tecnologia Fundamental

yago

WordNet

DBpedia

Justin Henry no cinema, como filho de
Meryl Streep ganhou nomeção ao Oscar"
e tentar retornar a resposta "Kramer vs.
humano faria.

Como resolver?

IBM Watson

"more than a quiz show"

The Grand Challenge

Em 2007, a IBM Resarch encarou o grande desafio de construir um sistema que pudesse responder, bem o suficente, pergutnas de domínio aberto para competir com campeões no jogo "Jeopardy!"

É um programa da televisão americana criado por Merv Griffin que apresenta a disputa entre 3 adversários, sendo vitorioso quem tiver mais acertos para perguntas sobre tópicos variados como história, literatura, arte, ciência, geografia, etc.

É um programa da televisão americana criado por Merv Griffin que apresenta a disputa entre 3 adversários, sendo vitorioso quem tiver mais acertos para perguntas sobre tópicos variados como história, literatura, arte, ciência, geografia, etc.

"Jeopardy! representa um grande desafio para um sistema de computação, devido à sua ampla gama de temas, a velocidade com que os concorrentes devem fornecer respostas precisas, e porque as pistas dadas aos participantes envolvem análise de significado sutil, ironia, charadas e outras complexidades de linguagem natural no qual os humanos são excelentes e os computadores tradicionalmente não são."

Os 3 desafios:

1) Perguntas de grande domínio: "Jeopardy!" questiona sobre centenas de milhares de coisas, usando expressões ricas e variadas em linguagem natural

2) Jogadores devem responder com precisão e convicção: Na média, campeões devem responder corretamente mais de 85% das questões que ele buzinou (há um dispositivo de botão para ser pressionado assim que o jogador acredita ter a resposta), e convicção suficiente para buzinar em 70% das perguntas

3) Respostas precisam ser rápidas: Vencedores devem rapidamente determinar sua convicção na resposta correta e buzinar suficientemente rápido para vencer os demais competidores

1) Perguntas de grande domínio: "Jeopardy!" questiona sobre centenas de milhares de coisas, usando expressões ricas e variadas em linguagem natural

2) Jogadores devem responder com precisão e convicção:

Na média, campeões devem responder corretamente mais de 85% das questões que ele buzinou (há um dispositivo de botão para ser pressionado assim que o jogador acredita ter a resposta), e convicção suficiente para buzinar em 70% das perguntas

3) Respostas precisam ser rápidas:

Vencedores devem rapidamente determinar sua convicção na resposta correta e buzinar suficientemente rápido para vencer os demais competidores

DeepQA

A Tecnologia Fundamental

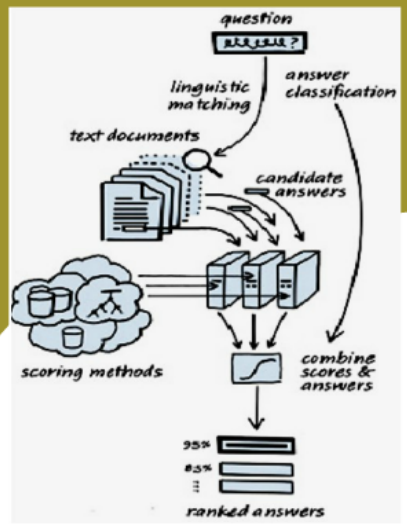
DeepQA:

Técnica para a geração de hipóteses, coleta maciça, de evidências, análise e correção.

Usada para desenvolver um sistema que supera o problema clássico de inteligência artificial (AI - Artificial Intelligence) de responder perguntas (QA - Question Answering).

Ideia básica geral:

Começar com documentos textuais, e depois construir um sistema para combinar, estatisticamente, as perguntas feitas com as respostas representadas nos documentos

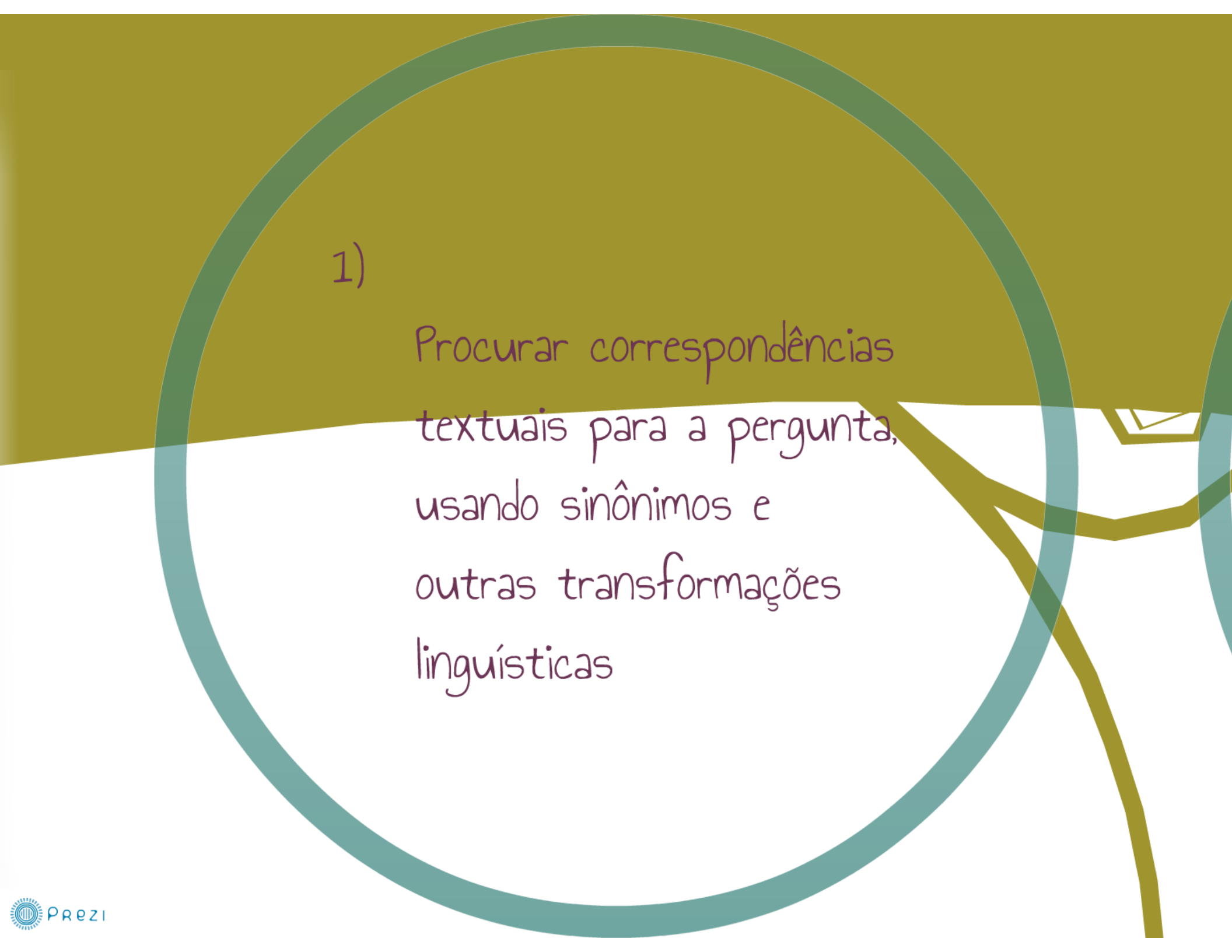


1) Procurar correspondências textuais para a pergunta, usando sinônimos e outras transformações linguísticas

2) Usar vários métodos diferentes para pontuar a lista de respostas possíveis, e combinar as pontuações para escolher a melhor resposta – o trabalho mais difícil

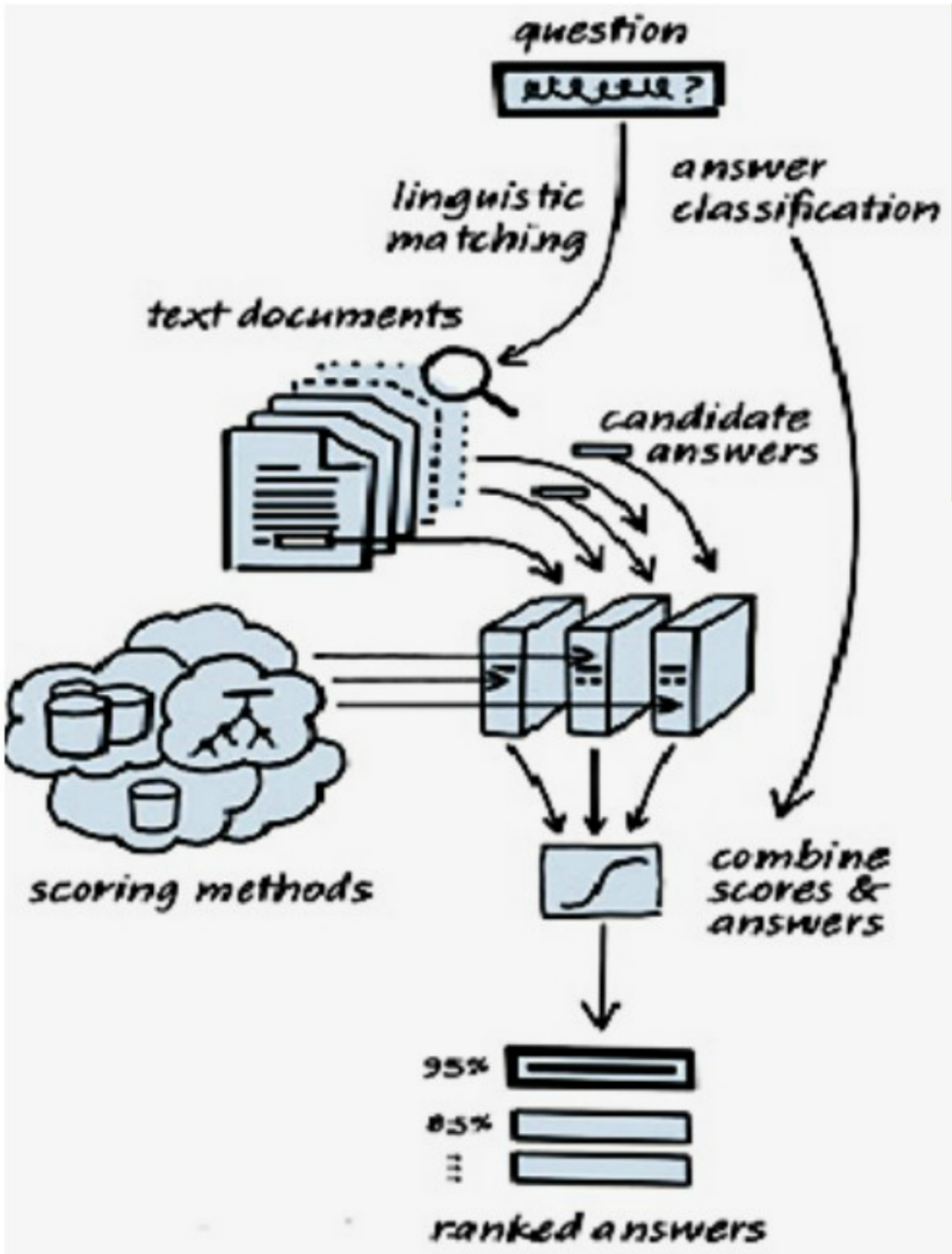
1)

Procurar correspondências textuais para a pergunta, usando sinônimos e outras transformações linguísticas

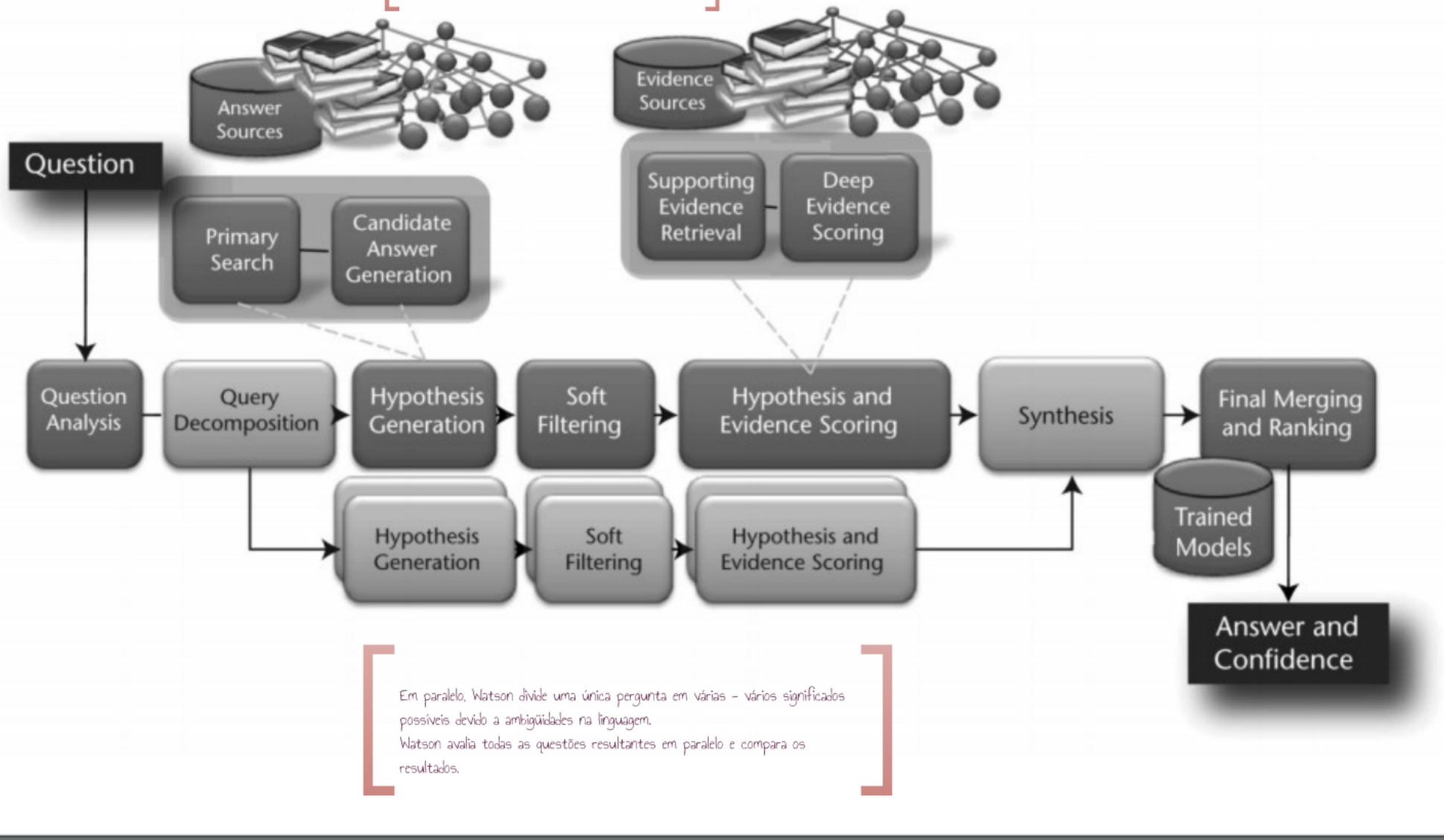


2)

Usar vários métodos diferentes para pontuar a lista de respostas possíveis, e combinar as pontuações para escolher a melhor resposta - o trabalho mais difícil


question
linguistic matching
answer classification
text documents
candidate answers
scoring methods
combine scores & answers
95%
85%
ranked answers

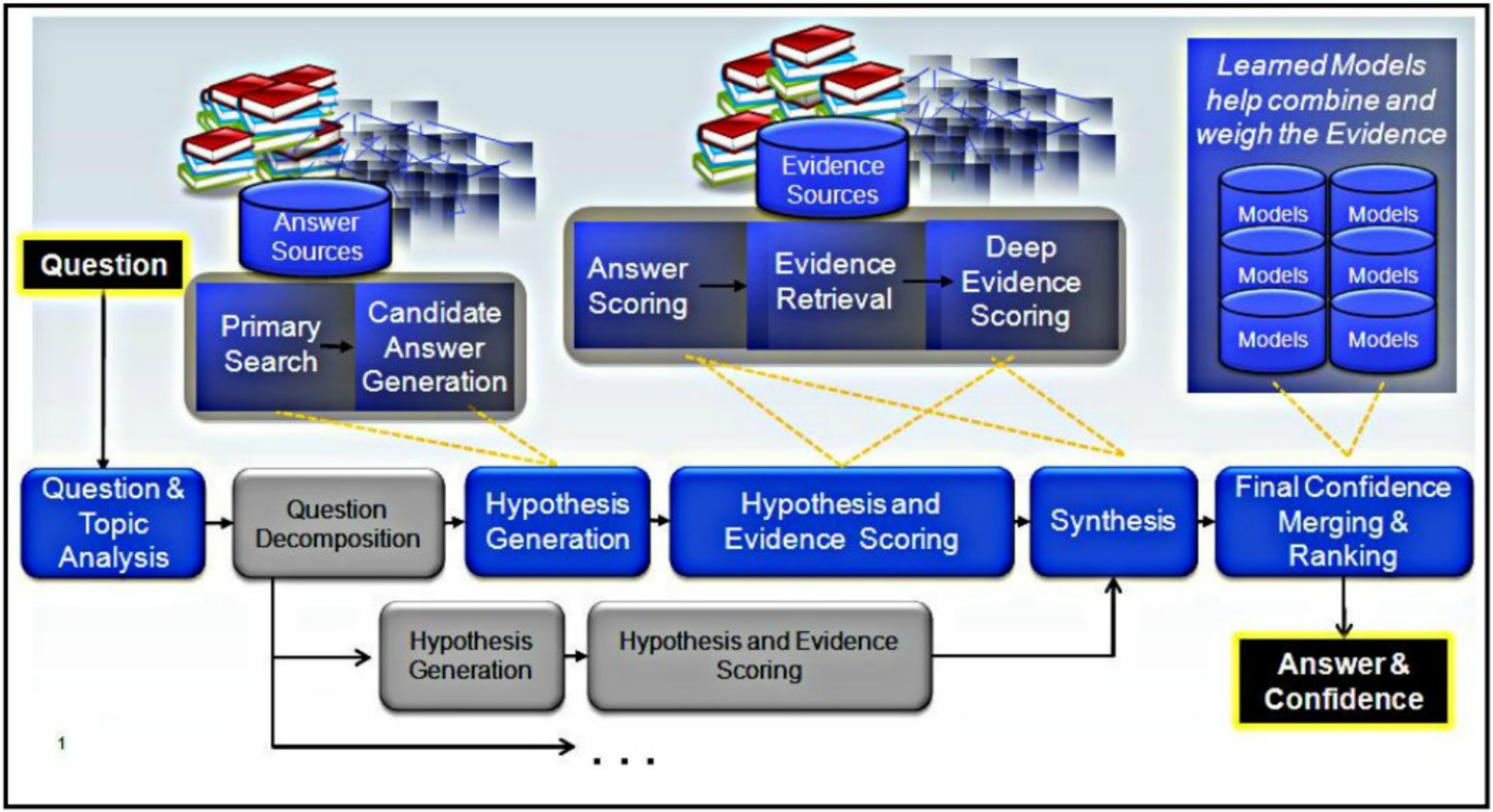
Sem conexão à Internet ou fonte de dados externa!



Em paralelo, Watson divide uma única pergunta em várias - vários significados possíveis devido a ambigüidades na linguagem.
Watson avalia todas as questões resultantes em paralelo e compara os resultados.

Em paralelo, Watson divide uma única pergunta em várias - vários significados possíveis devido a ambigüidades na linguagem.
Watson avalia todas as questões resultantes em paralelo e compara os resultados.

Sem conexão à Internet ou fonte de dados externa!


Question
Answer Sources
Primary Search
Candidate Answer Generation
Evidence Sources
Answer Scoring
Evidence Retrieval
Deep Evidence Scoring
Learned Models help combine and weigh the Evidence
Models
Models
Models
Models
Models
Models
Question & Topic Analysis
Question Decomposition
Hypothesis Generation
Hypothesis and Evidence Scoring
Synthesis
Final Confidence Merging & Ranking
Hypothesis Generation
Hypothesis and Evidence Scoring
Answer & Confidence
1
. . .

Construção do Conhecimento:

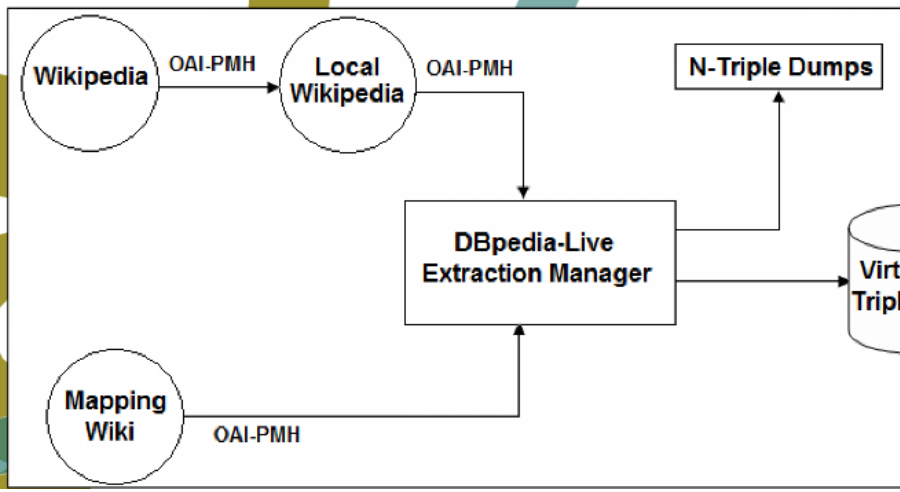
Antes que qualquer pergunta seja respondida, Watson precisa obter conhecimento.

Não é um processo em tempo real, e é feito antes, e de forma independente, do processo pergunta-resposta.

A IBM não revela a real origem dos dados para o Jeopardy!, mas, de acordo com algumas entrevistas com a equipe, especula-se que são usadas várias enciclopédias, dicionários, tesauros, artigos em rede de notícias, obras literárias, dbPedia, WordNet, e a ontologia Yago. Contudo, não é possível encontrar relatório explicando como esse conteúdo é organizado e usado.

DBpedia

Informações estruturadas a partir da Wikipedia, disponibilizadas na Web, permitindo consultas sofisticadas.
É um esforço comunitário.
Considerada a Web Semântica espelhada da Wikipedia.



WordNet
A lexical database for English

É um grande banco de dados léxico de inglês. Substantivos, verbos, adjetivos e advérbios são agrupados em conjuntos de sinônimos cognitivos (synsets), cada um expressando um conceito distinto. Synsets são interligados por meio de relações léxicas e semântico-conceituais.

O resultado é uma rede de palavras e conceitos, significativamente relacionados, que pode ser navegada com o browser.

YAGO

select knowledge

É uma grande base de conhecimento semântica, sobre mais de 10 milhões de entidades (pessoas, organizações, cidades, etc.), contendo mais de 120 milhões de fatos sobre estas.
A precisão foi avaliada manualmente, confirmando 95%.
É uma ontologia fiel ao tempo e ao espaço - atribui dimensão temporal e espacial para muitos de seus fatos e entidades.
Tem domínios temáticos, do WordNet como "música" e "ciência".

Análise da Pergunta:

Watson faz uso de algumas técnicas de processamento da linguagem natural (NLP - Natural Language Processing).

As perguntas são primeiro analisadas sintaticamente e então semanticamente, para obter um formato lógico.

São identificados os papéis semânticos, podendo identificar o tipo de sujeito e objeto.

Algumas técnicas de aprendizado de máquina são utilizadas.

Quanto mais Watson lê, mais aprende como a linguagem é usada - contexto e associações de palavras com o uso de modelos de linguagem.

Se o nível de confiança é baixo, Watson acredita que há um trocadilho ou algo não foi compreendido.
Se o nível de confiança, por outro lado, é alto, Watson acredita que há uma boa chance de acertar a pergunta.

Nesse ponto, Watson também determina o tipo de resposta léxica eperada (LAT - Lexical Answer Type) - palavra ou substantivo que especifica o tipo de resposta mesmo sem entender a semântica.

Geração de Hipótese

Com o resultado da Análise da Pergunta, é a vez de gerar a resposta candidata (hipótese). executando uma busca inicial pelas fontes de respostas (a Base de Conhecimento já vista) com vários motores de busca em diferentes abordagens básicas e SPARQL.

Posteriormente, o resultado da pesquisa é usado para gerar as respostas candidatas que serão pontuadas e comparadas.

A criação das hipóteses depende do tipo de dados.

Por exemplo:

- Uma resposta a partir de um trecho de texto (não-estruturados) poderá vir de técnicas de reconhecimento de entidades nomeadas
- Uma resposta a partir de um banco de dados relacional (estruturada) poderá ser o resultado exato da consulta

A resposta correta não é gerada nesta fase. O sistema não tem esperança de responder à pergunta agora!
Esta etapa favorece significativamente a precisão, com a expectativa de que o resto do processamento destrinche a resposta correta, mesmo se o conjunto de candidatas é muito grande.
Watson está configurado para retornar cerca de 250 respostas candidatos durante esta fase. Serão cerca de 100 após a "filtragem soft", com filtros simples, que determina a probabilidade da resposta de ser da LAT (determinado na fase de análise) correta.

A "Caixa Preta" - Patente

O problema clássico de QA:

"Primeiro papel de Justin Henry no cinema, como filho de Dustin Hoffman e Merryl Streep ganhou nomeção ao Oscar"

O computador deve tentar retornar a resposta "Kramer vs. Kramer" como um humano faria.

Como resolver?

80% dos dados mundiais são desestruturados!

É razoável que se use técnicas de NLP (Natural Language Processing)

A maioria das empresas processam dados estruturados em seus negócios e os armazenam em banco de dados sem transferí-los para dados desestruturados.

Para apoiar QA com dados estruturados, novas técnicas precisam ser desenvolvidas, por exemplo NLDB (Natural Language DataBase).

Além da base de dados, há muitos novos dados estruturados, como por exemplo RDF e linguagens de pesquisa semântica, tais como SPARQL.

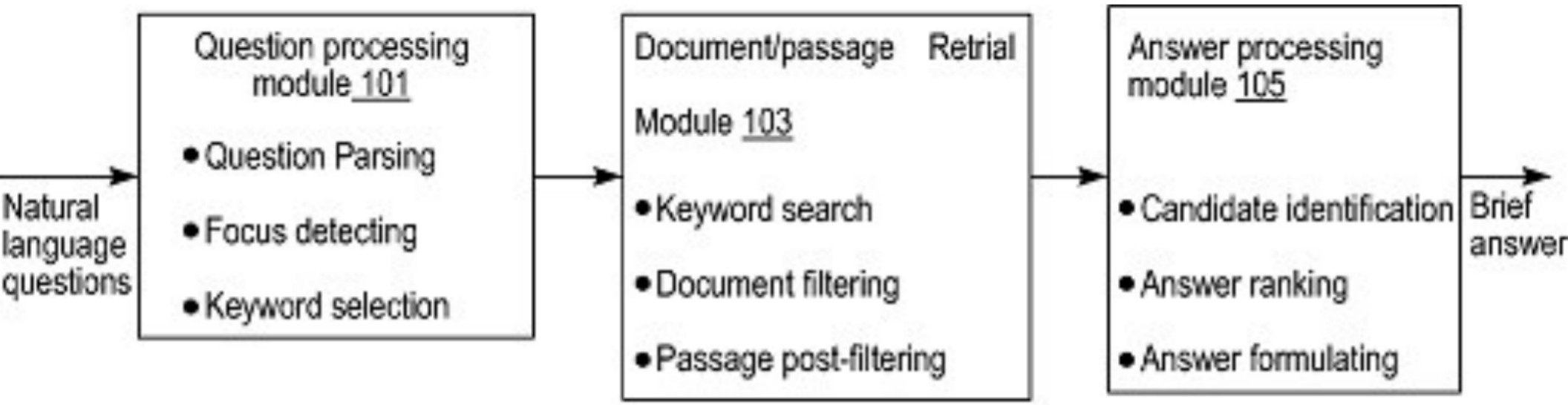

Natural language questions
Question processing module 101
• Question Parsing
• Focus detecting
• Keyword selection
Document/passage Retrial Module 103
• Keyword search
• Document filtering
• Passage post-filtering
Answer processing module 105
• Candidate identification
• Answer ranking
• Answer formulating
Brief answer

FIG. 1

O método implementado para selecionar respostas a perguntas em linguagem natural:

- Detectar entidade
- Extrair informação relacionada a uma resposta
- Procura em linked data de acordo com a entidade
- Gerar pelo menos uma resposta candidata
- Analisar a resposta candidata de acordo com a informação relacionada e obter um valor de uma característica da resposta candidata
- Avaliar cada resposta candidata através da sintetização do valor de característica

Procura em linked data:

- Busca a URI (Uniform Resource Indentifier), fazendo a correspondência com a entidade nomeada
- Ampliação da busca pelas URIs ligadas à primeira URI, sempre de acordo com a entidade nomeada
- Geração da resposta candidata de acordo com a URI

Resp
linked
com
sinte
carac
avalia

Respostas candidatas em diferentes linked data são mescladas de acordo com a característica das respostas, sintetizando o valor das características antes mesmo de avaliar a resposta

Mais adiante é adicionado aprendizado de máquina de acordo com a característica da resposta para treinar um modelo de pontuação

Como exemplo na patente, optaram por usar RDF. Particularmente o LOD (Linked Open Data) da W3C, com mais de 30 conjuntos de dados, consistindo em mais de 2 bilhões de triplas.
Uma visão de RDF virtual pode ser convenientemente construída usando ferramentas de Web semântica, tais como Virtuoso, D2R e SeDA.

[0034] Besides the physical RDF data, virtual RDF datasets are growing as well. Many large corporations manage and process structured data inside their individual business system and need to integrate their structured data as well. A virtual RDF view can be conveniently built based on their structured databases using some semantic Web tools such as Virtuso, D2R and SeDA.

[0035] However, it should be understood by those skilled in the art that, the present invention is not limited to RDF data, but can also be applied to various linked data, such as linked data obtained by mapping Micro-format data.

[0036] Next, Dbpedia is used as a particular example of RDF, and the principle of the present invention is illustrated hereinafter by describing how the answer to the natural language question "In this 1992 Robert Altman film, Tim Robbins gets angry messages from a screenwriter he's snubbed" is obtained.

[0037] Some RDF triple data related to the above natural language question in DBpedia are listed below first, and its graph structure is illustrated in FIG. **2**.

- <http://dbpedia.org/resource/The_Player>
  <http://dbpedia.org/property/director>
  <http://dbpedia.org/resource/Robert_Altman>.
- <http://dbpedia.org/resource/The_Player>
  <http://www.w3.org/2000/01/rdf-schema%23label>
  "The Player"@en.
- <http://dbpedia.org/resource/Gosford_Park>
  <http://dbpedia.org/property/director>
  <http://dbpedia.org/resource/Robert_Altman>.
- <http://dbpedia.org/resource/Robert_Altman>
  <http://dbpedia.org/property/birthPlace>
  <http://dbpedia.org/resource/Kansas_City%2C_Missouri>.
- <http://dbpedia.org/resource/The_Player>
  <http://dbpedia.org/property/starring>
  <http://dbpedia.org/resource/Tim_Robbins>.
- <http://dbpedia.org/resource/Tim_Robbins>
  <http://dbpedia.org/property/spouse>
  <http://dbpedia.org/resource/Susan_Sarandon>.
- <http://dbpedia.org/resource/The_Player>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
  <http://dbpedia.org/class/yago/MotionPictureFilm103789400>.
- <http://dbpedia.org/class/yago/MotionPictureFilm103789400>
  <http://www.w3.org/2000/01/rdf-schema%23subClassOf>
  <http://dbpedia.org/class/yago/Film103435300>.
- <http://dbpedia.org/class/yago/Film103435300>
  <http://www.w3.org/2000/01/rdf-schema%23label>
  "Film"@en.

[0038] In FIG. **2**, circles represent URIs (Universal Resource Identifiers) related to named entities, which are subjects and objects in the RDF triples. Lines connecting two circles indicate the relationship between the named entities, which are predicates in the RDF triples. Taking the first of the above RDF triples as an example: "<http://dbpedia.org/resource/The_Player> <http://dbpedia.org/property/director> <http://dbpedia.org/resource/Robert_Altman>", "The_Player" and "Robert_Altman" are named entities, "<http://dbpedia.org/resource/The_Player>" is the URI related to the named entity "The_Player", and <http://dbpedia.org/resource/Robert_Altman>" is the URI related to the named entity "Robert_Altman", which are therefore indicated by circles in the graph structure shown in FIG. **2**. Furthermore, as the predicate in the RDF triple, "<http://dbpedia.org/property/director>" indicates the relationship between the named entities "The_Player" and "Robert_Altman", that is, "Robert_Altman" is the "director" of the film "The_Player". Other RDF triples can be parsed in the same way and are not listed here one by one.

[0039] FIG. **3** is a general flow chart of a method for processing natural language questions according to an embodiment of the present invention. As shown in FIG. **3**, the method for processing natural language questions according to an embodiment of the present invention includes named entity detection step **S301**, answer-related information extraction step **S303**, linked database retrieval step **S305**, candidate answer generation step **S307**, feature value obtaining step **S309**, and candidate answer evaluation step **S311**.

[0040] First, in step **S301** for named entity detection, a natural language question inputted by the user is parsed and a named entity is detected. Next, information related to an answer is extracted from the natural language question in the answer-related information extraction step **S303**.

[0041] For example, from the natural language question "In this 1992 Robert Altman film, Tim Robbins gets angry messages from a screenwriter he's snubbed", named entities "Robert_Altman" and "Tim Robbins" can be detected, and information "film" related to the type of the answer and time verification information "1992" related to the answer can be extracted.

[0042] Then, in step **S305** for linked database retrieval, search is performed in different data sources such as linked data of DBpedia and IMDB based on the named entities detected in named entity detection step **S301**. Next, in candidate answer generation step **S307**, a candidate answer is generated based on a search result from linked database retrieval step **S305**.

[0043] FIG. **4** is a flow chart of the step of searching in a linked database and generating a candidate answer according to an embodiment of the present invention. As shown in FIG. **4**, first in matching step **S401**, a URI matching to a named entity is searched for in linked data based on similarity. For the above exemplary natural language question, based on the named entities "Robert_Altman" and "Tim Robbins" detected in named entity detection step **S301**, matching URIs "<http://dbpedia.org/resource/Robert_Altman>" and "<http://dbpedia.org/resource/Tim_Robbins>" can be retrieved from DBpedia respectively.

[0044] Next, in spreading activation step **S403**, a URI directly linked to the URI matching to the named entity is searched for with spreading activation using linking relationship between URIs. In the example above, for the URI "<http://dbpedia.org/resource/Robert_Altman>" matching to the named entity "Robert_Altman", URIs directly linked to it can be obtained easily from the graph structure shown in FIG. **2** by spreading activation, e.g. "<http://dbpedia.org/resource/The_Player>", "<http://dbpedia.org/resource/Gosford_Park>" and "<http://dbpedia.org/resource/Kansas_City%2C_Missouri>". For the URI "<http://dbpedia.org/resource/Tim_Robbins>" matching to the named entity "Tim Robbins", URIs directly linked to it can also be easily obtained from the graph structure shown in FIG. **2** by spreading activation, e.g. "<http://dbpedia.org/resource/The_Player>" and "<http://dbpedia.org/resource/Susan_Sarandon>".

[0045] After obtaining each of the above URIs by spreading activation, candidate answers can be extracted from the directly linked URIs, where the candidate answers may be a label contained in a URI. For the above example, candidate answers such as "The_Player", "Gosford_Park", "Kansas_

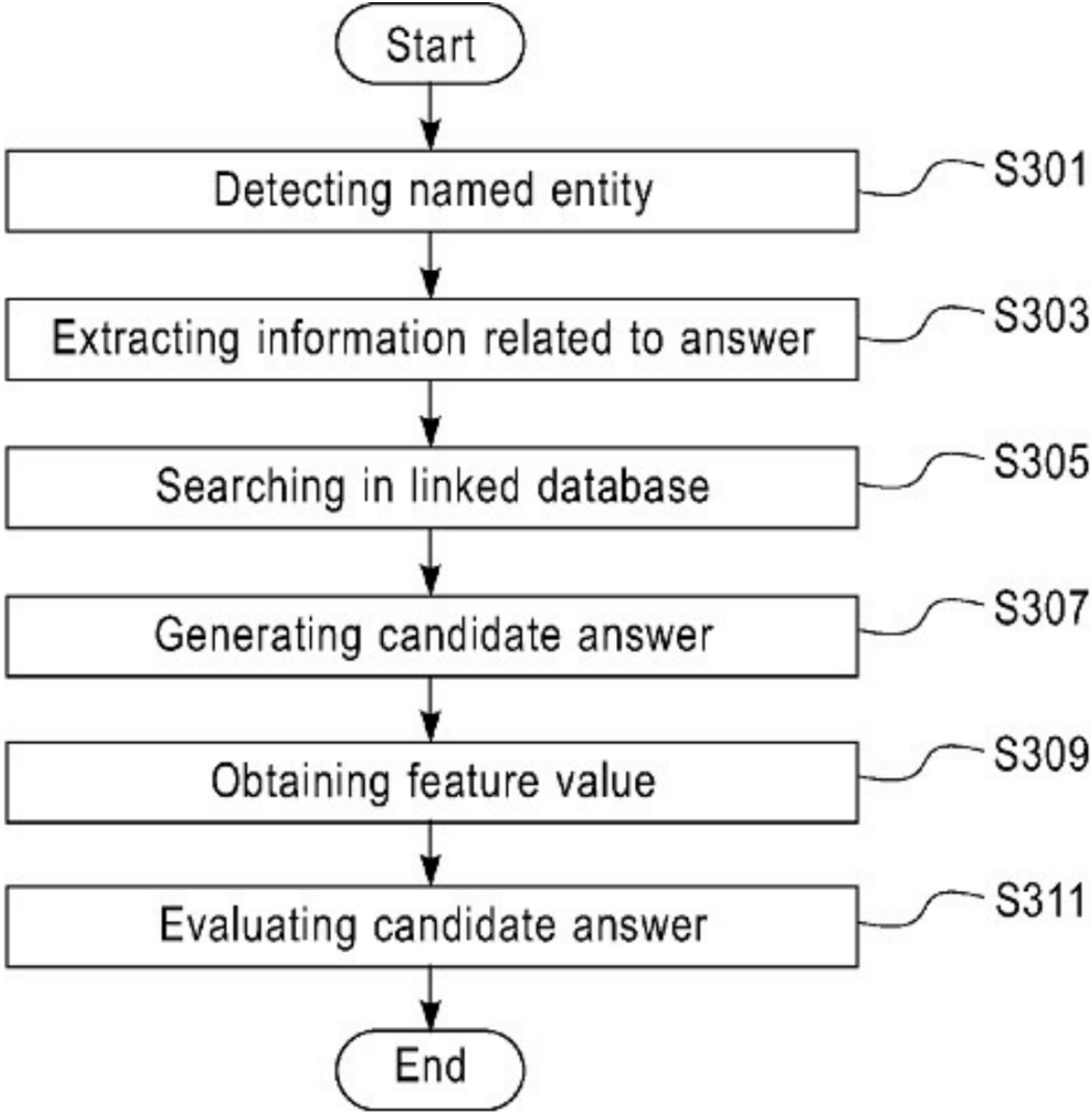

Start
Detecting named entity
S301
Extracting information related to answer
S303
Searching in linked database
S305
Generating candidate answer
S307
Obtaining feature value
S309
Evaluating candidate answer
S311
End

FIG. 3

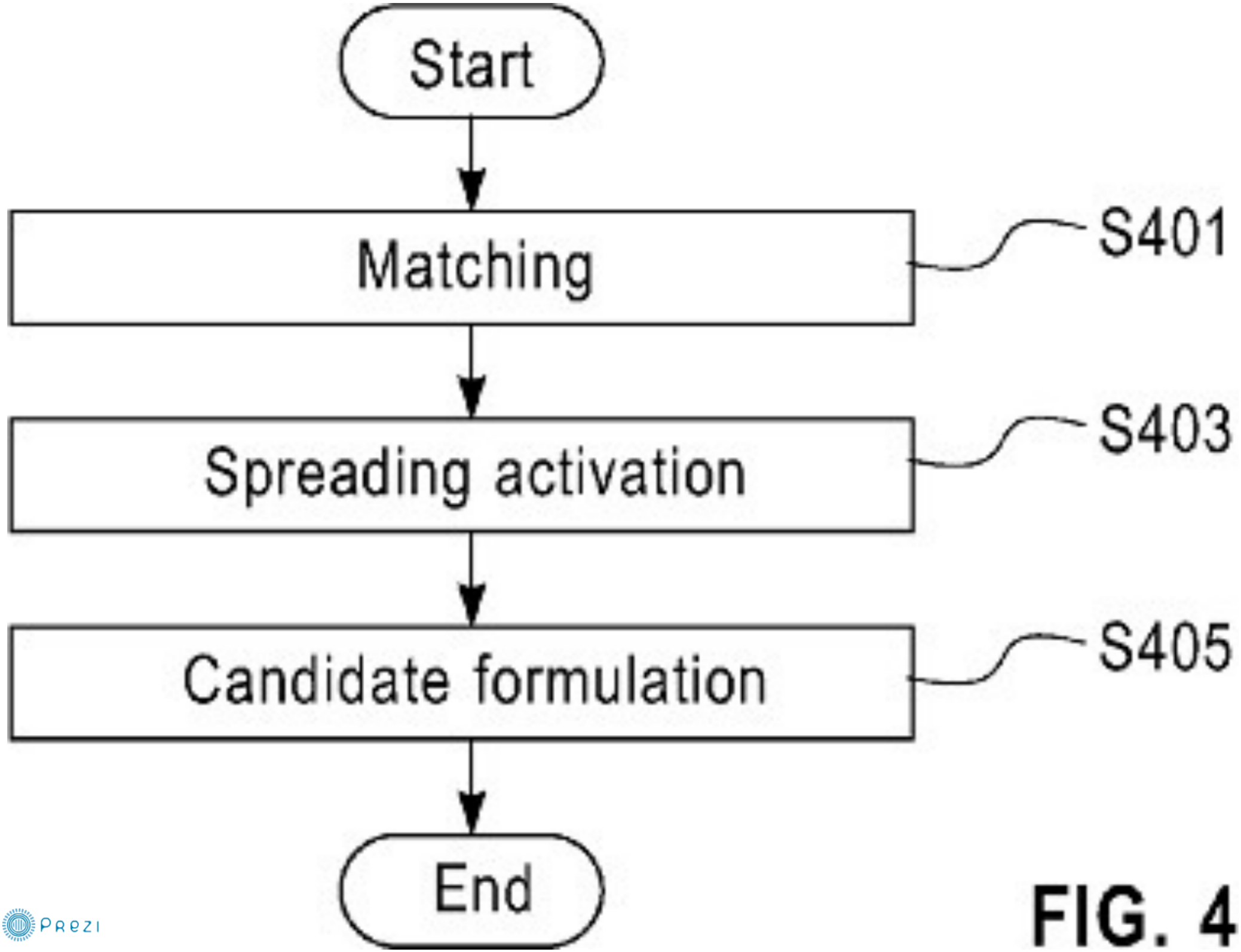

Start
Matching
S401
Spreading activation
S403
Candidate formulation
S405
End

FIG. 4

City" and "Susan_Sarandon" can be extracted from the directly linked URIs obtained in spreading activation step **S403**. In this embodiment, URIs directly linked to the URIs matching to the named entities are searched for with spreading activation, and candidate answers are generated based on directly linked URIs. However, those skilled in the art would understand that, it is not limited to the directly linked URIs in searching with spreading activation and generating candidate answers.

[0046] After generating candidate answers according to the process illustrated in FIG. **4**, next in the feature value obtaining step **S309** shown in FIG. **3**, the candidate answers are parsed based on the information related to the answer extracted in answer-related information extraction step **S303**, so as to obtain values of a feature of the candidate answers.

[0047] The feature of the candidate answers here includes the information related to the answer, and the number of directly linked URIs associated with the candidate answer. The information related to the answer includes, for example, information "film" related to the type of the answer and time verification information "1992" related to the answer. Answer-type related information may be indicated by "tycor", and time verification information may be directly indicated by "year". The number of directly linked URIs associated with the candidate answer is, for example, the number of URIs directly linked to each of the URIs of the candidate answers, which feature is hereby indicated by "triple". Accordingly, values of features of each candidate answers for the above specific example are given in Table 1.

TABLE 1

Feature values of candidate answers

Candidate answer	Feature value		
The_Player	tycor = 1	triple = 2	year = 1
Gosford_Park	tycor = 1	triple = 1	year = 0
Kansas_City	tycor = 0	triple = 1	year = 0
Susan_Sarandon	tycor = 0	triple = 1	year = 0

[0048] As can be seen from Table 1, for the feature "tycor", as the candidate answers "The_Player" and "Gosford_Park" both are film titles, consistent with the answer-type related information "film" extracted in answer-related information extraction step **S303**, therefore their tycor=1. The candidate answer "Kansas_City" is a city name, and "Susan_Sarandon" is a human name, not consistent with the answer-type related information "film", therefore their tycor=0. For the feature "triple", it can be seen intuitively from FIG. **2** that, the numbers of URIs directly linked to the candidate answers "The_Player", "Gosford_Park", "Kansas_City" and "Susan_Sarandon" and related to the named entities "Robert_Altman" and "Tim Robbins" are 2, 1, 1 and 1, respectively, therefore their "triple" values are assigned with 2, 1, 1 and 1, respectively. For the feature "year", as the time verification information "1992" extracted in answer-related information extraction step **S303** is present only in the URI "<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>" linked to the candidate answer "The_Player", the "year" value of the candidate answer "The_Player" is assigned with 1, and the "year" value of the other candidate answers are assigned with zero.

[0049] It should be noted that, features of the candidate answers are not limited to information related to a type of the answer, the number of directly linked URIs related to the candidate answers, and time verification information related to the answer as mentioned in the above example, but may include various information relating to an answer, named entity, URI or the like, for example, linking information between URIs matching to a named entity.

[0050] After obtaining the values of features of candidate answers in feature value obtaining step **S309**, the values of features of candidate answers can be synthesized in the candidate answer evaluation step **S311**, so that each of the candidate answers can be evaluated and a best answer can be provided to the user.

[0051] ~~According to a preferred embodiment of the present~~ invention, machine learning is performed in advance in accordance with given features of candidate answers, to obtain a satisfying scoring model. Accordingly, when synthesizing the values of features of candidate answers in the candidate answer evaluation step **S311**, a score can be computed for each candidate answer using the trained scoring model, and the candidate answer with the highest score can be selected as the final answer provided to the user. Table 2 below shows the scoring results obtained by evaluating each candidate answer in the above example.

TABLE 2

Evaluation of candidate answers

Candidate answer	Feature value			Evaluation score
The_Player	tycor = 1	triple = 2	year = 1	100
Gosford_Park	tycor = 1	triple = 1	year = 0	60
Kansas_City	tycor = 0	triple = 1	year = 0	0
Susan_Sarandon	tycor = 0	triple = 1	year = 0	0

[0052] In Table 2, for the candidate answer "The_Player", not only its answer type matches the desired answer type, but also its time related verification information conforms, and it has the largest number of directly linked URIs associated with candidate answers, therefore, it is given the highest score **100** and provided to the user as the best answer. For the candidate answer "Gosford_Park", as its feature "year=0", and the number of directly linked URIs associated with candidate answers is only 1, therefore it is not the best answer and given a score **60** although its answer type matches the desired answer type. Furthermore, for the candidate answers "Kansas_City" and "Susan_Sarandon", as both of their answer-type values are 0 and do not match the desired answer type, both of their final evaluation scores are **0**.

[0053] As a matter of course, the evaluation results in Table 2 are given as examples only. In practice, different weights may be given to the features based on different situations, and evaluation of candidate answers can be performed accordingly.

[0054] Furthermore, it should also be noted that, candidate answers are not necessarily obtained from the same linked data, e.g. DBpedia used in the above example. Candidate answers may be retrieved from different linked data. Therefore, if candidate answers are obtained from different linked data respectively, before evaluating the candidate answers in candidate answer evaluation step **S311**, the candidate answers retrieved from different linked data may be merged according to a feature of the candidate answers, so that repeated candidate answers can be avoided.

[0055] The processing process of the method for processing natural language questions according to an embodiment of the present invention is described above. The working

principle of an apparatus for processing natural language questions according to an embodiment of the present invention is described hereinafter in conjunction with FIG. **5** and FIG. **6**.

[0056] FIG. **5** is a structural block diagram of an apparatus **500** for processing natural language questions according to an embodiment of the present invention. As shown in FIG. **5**, the apparatus **500** for processing natural language questions according to an embodiment of the present invention includes: a question parsing module **501**, a candidate answer generating module **503**, a feature value generating module **505**, and a candidate answer evaluating module **507**.

[0057] First, the question parsing module **501** parses the natural language question, detects a named entity and extracts information related to an answer from the natural language question. Then, the candidate answer generating module **503** searches in linked data such as DBpedia and IMDB according to the named entity detected by the question parsing module **501**, and thereby generates candidate answers. Next, the feature value generating module **505** parses the candidate answers generated by the candidate answer generating module **503** according to the information related to the answers, and obtains values of a feature of the candidate answers. Finally, the candidate answer evaluating module **507** evaluates each candidate answer by synthesizing the values of the features of the candidate answers, and provides the best candidate answer to the user as the final result.

[0058] FIG. **6** is a schematic structural block diagram of a candidate answer generating module **600** according to a preferred embodiment of the present invention. As shown in FIG. **6**, the candidate answer generating module **600** according to the embodiment includes a matching unit **601**, a spreading activation unit **603** and a candidate generating unit **605**.

[0059] The matching unit **601** searches for a URI matching to the named entity in the linked data based on similarity; the spreading activation unit **603** searches with spreading activation for a URI directly linked to the URI obtained by the matching unit **601** matching to the named entity by using the linking relationship between URIs; and the candidate generating unit **605** generates the candidate answers according to the directly linked URI retrieved by the spreading activation unit **603**.

[0060] The candidate generating unit **605** may use a label contained in a URI as a candidate answer. The feature of the candidate answers should at least include information related to the answer, and the number of directly linked URIs associated with the candidate answers. The information related to the answer at least includes the type of the answer.

[0061] According to a preferred embodiment of the present invention, the information related to the answer may further include time verification information related to the answer extracted from the natural language question, and the features of the candidate answers may further include linking information between URIs matching to a named entity.

[0062] It should be noted that, candidate answers are not necessarily obtained from the same linked data, but may be retrieved from different linked data. Therefore, a preferred embodiment of the present invention may include a merging module (not shown in the figure), which is configured to, if candidate answers are obtained from different linked data, merge the candidate answers retrieved from different linked data according to a feature of the candidate answers before the candidate answer evaluation module **507** evaluates the candidate answers, so that repeated candidate answers can be avoided.

[0063] In addition, the apparatus for processing natural language questions according to a preferred embodiment of the present invention may further include a training module (not shown in the figure), which is configured to perform machine learning in advance according to given features of candidate answers, so as to obtain a satisfying scoring model. Accordingly, when the candidate evaluation module **507** synthesizes the values of features of candidate answers, a score can be computed for each candidate answer using the trained scoring model, and the candidate answer with the highest score can be selected as the final answer provided to the user.

[0064] It should also be noted that, detailed processing processes of the question parsing module **501**, the candidate answer generating module **503**, the feature value generating module **505**, and the candidate answer evaluating module **507** in the apparatus for processing natural language questions according to the present invention are similar to those of named entity detection step S**301**, answer-related information extraction step S**303**, linked database retrieval step S**305**, candidate answer generation step S**307**, feature value obtaining step S**309** and candidate answer evaluation step S**311** in the method for processing natural language questions described with reference to FIG. **3**, respectively. And detailed processes of the matching unit **601**, the spreading activation unit **603** and the candidate generating unit **605** in the candidate answer generating module **600** are similar to those of the matching step S**401**, the spreading activation step S**403** and the candidate generating step S**405** in the candidate answer generation method described with reference to FIG. **4**. Therefore, further detailed description is omitted.

[0065] As can be seen from the description of the embodiments of the present invention and the analysis of the prior art, when analyzing documents/sentences/words using NLP techniques, as natural language is extremely hard to be well parsed, existing QA systems over unstructured data have to process many ambiguous problems. However, the method and apparatus for processing natural language questions according to an embodiment of the present invention is a QA system over structured data, therefore may improve precision of QA systems based on existing huge amount of linked data.

[0066] In addition, the method and the apparatus for processing natural language questions according to an embodiment of the present invention may assist corporations enable QA systems over a virtual RDF view, applicable for huge amount of RDF data and virtual RDF data generated by the corporations without need of changing the existing QA systems.

[0067] The basic principle of the present invention is described in conjunction with the embodiments above. However, for those skilled in the art, it should be understood that, each or any step or component of the method and the apparatus of the present invention may be implemented with hardware, firmware, software or a combination thereof in any computing apparatus, including processors, storage medium and the like, or a network of computing apparatuses, which can be done by those skilled in the art with basic programming skills after reading the specification of the present invention.

[0068] Therefore, the object of the present invention may also be implemented by executing a program or a series of programs on any computing apparatus. The computing appa-

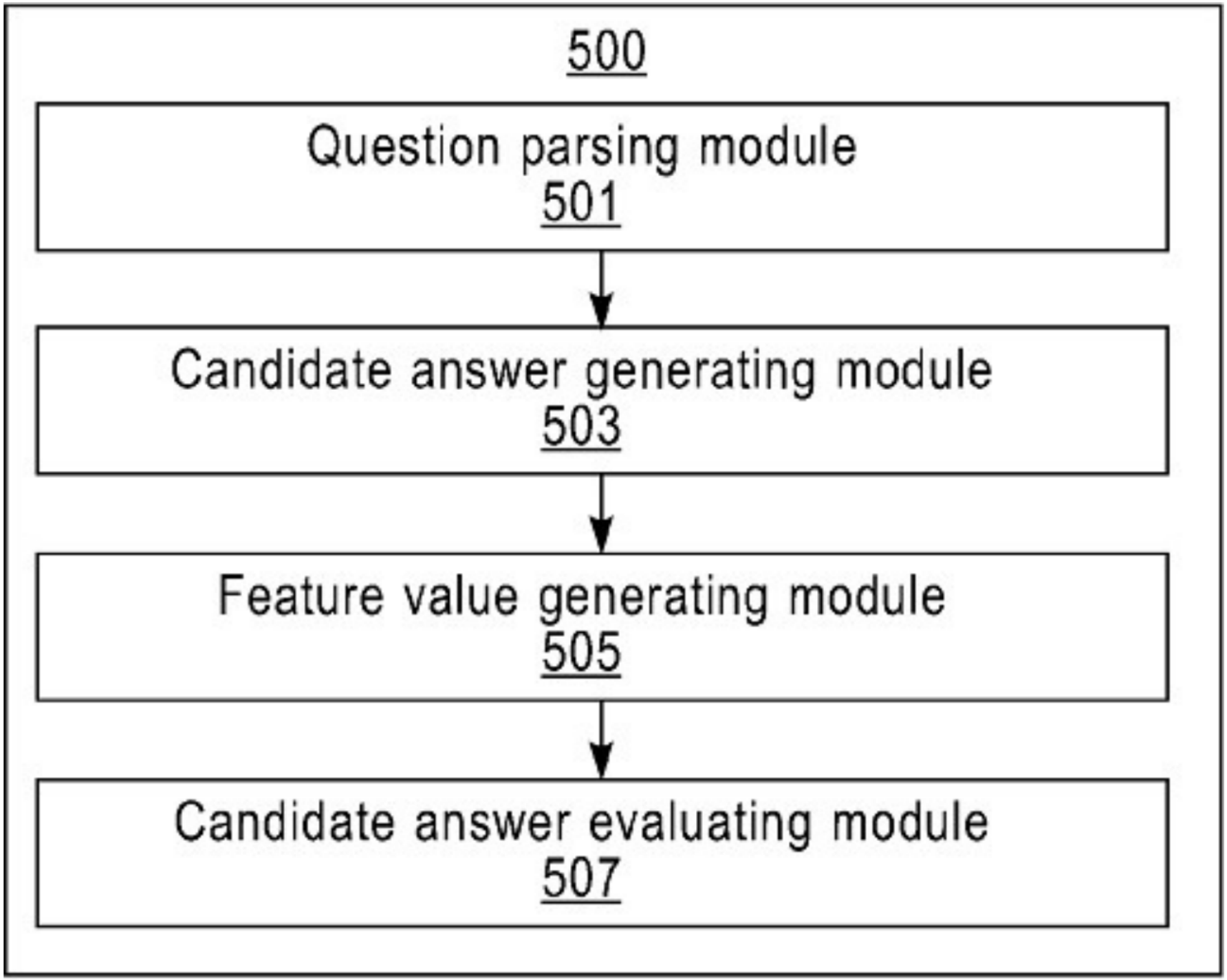

500
Question parsing module
501
Candidate answer generating module
503
Feature value generating module
505
Candidate answer evaluating module
507

FIG. 5

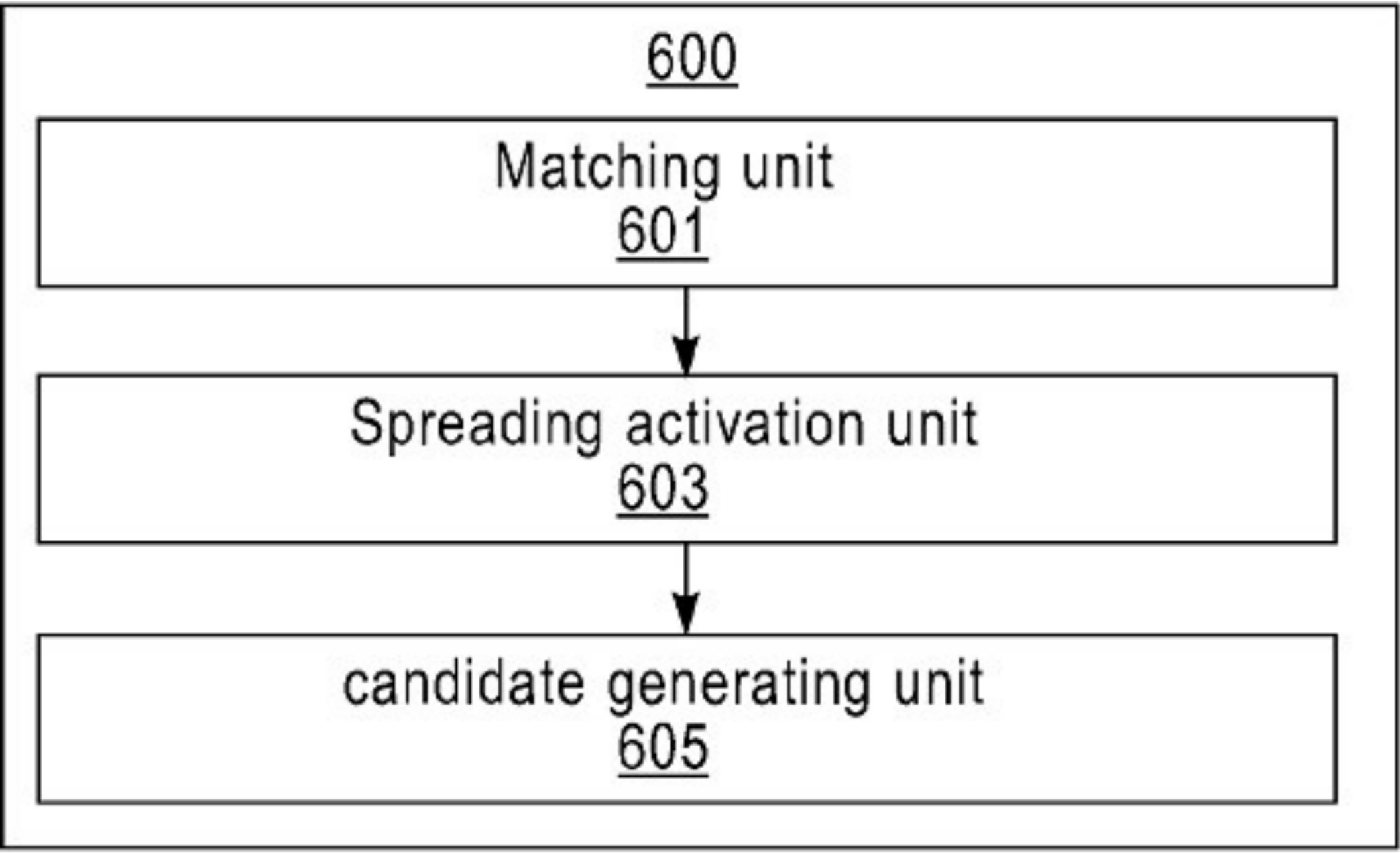

600
Matching unit
601
Spreading activation unit
603
candidate generating unit
605

FIG. 6

Referências

Li, Yang. "A Summary about Watson and Jeopardy."

Ferrucci, David, et al. "Building Watson: An overview of the DeepQA project." AI Magazine 31.3 (2010): 59-79.

Lally, Adam, and Paul Fodor. "Natural Language Processing With Prolog in the IBM Watson System." Retrieved June 15 (2011): 2011.

Ferrucci, David, et al. "Watson: Beyond Jeopardy!." Artificial Intelligence (2012).

Patentes:

- US 2010/0299139 A1 (23/04/2010)
- US 2012/0078837 A1 (31/03/2011)
- US 2012/0077178 A1 (24/09/2011)
- US 2012/0078826 A1 (28/09/2011)