US 20170169105A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2017/0169105 A1

ANDRADE SILVA et al. (43) Pub. Date: Jun. 15, 2017

(54) DOCUMENT CLASSIFICATION METHOD

(71) Applicant: **NEC Corporation**, Tokyo (JP)

(72) Inventors: **Daniel Georg ANDRADE SILVA**, Tokyo (JP); **Hironori MIZUGUCHI**, Tokyo (JP); **Kai ISHIKAWA**, Tokyo (JP)

(73) Assignee: **NEC Corporation**, Tokyo (JP)

(21) Appl. No.: **15/039,347**

(22) PCT Filed: **Nov. 27, 2013**

(86) PCT No.: **PCT/JP2013/082515**

§ 371 (c)(1),
(2) Date: **May 25, 2016**

Publication Classification

(51) **Int. Cl.**

G06F 17/30	(2006.01)
G06F 17/18	(2006.01)
G06N 99/00	(2006.01)
G06F 17/11	(2006.01)

(52) **U.S. Cl.**

CPC .. ***G06F 17/30705*** (2013.01); ***G06F 17/30663*** (2013.01); ***G06F 17/30011*** (2013.01); ***G06F 17/11*** (2013.01); ***G06F 17/18*** (2013.01); ***G06N 99/005*** (2013.01)

(57) **ABSTRACT**

A document classification method includes a first step for calculating smoothing weights for each word and a fixed class, a second step for calculating smoothed second-order word probability, and a third step for classifying document including calculating the probability that the document belongs to the fixed class.

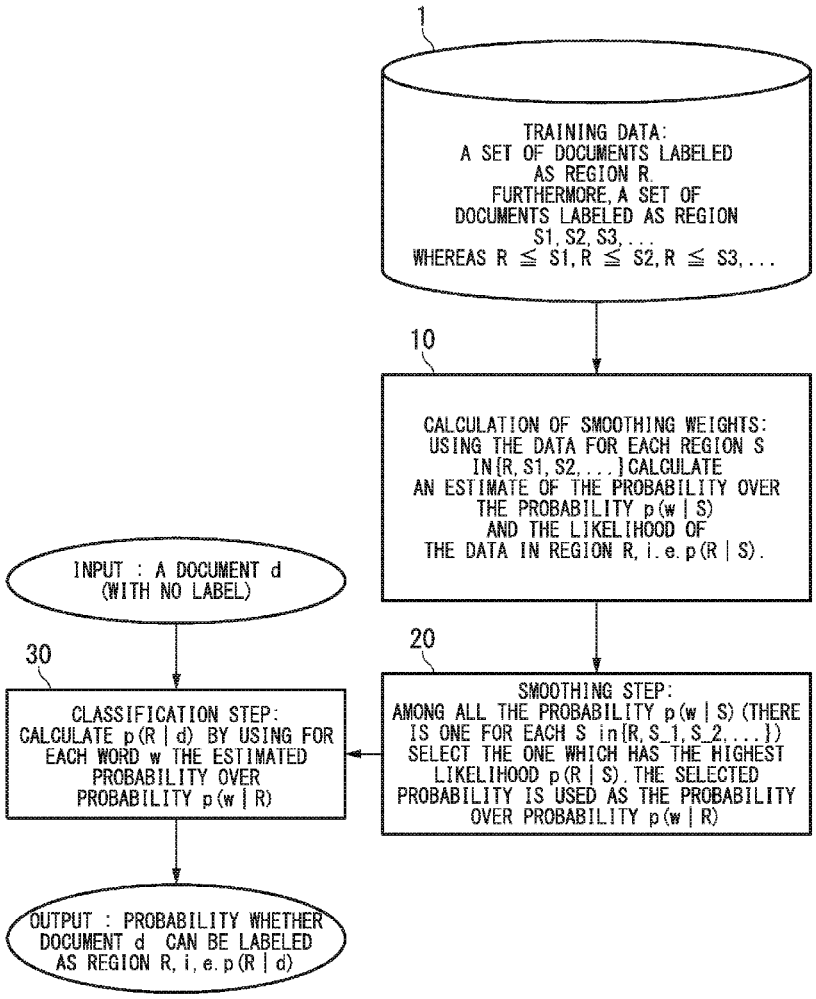


FIG. 1

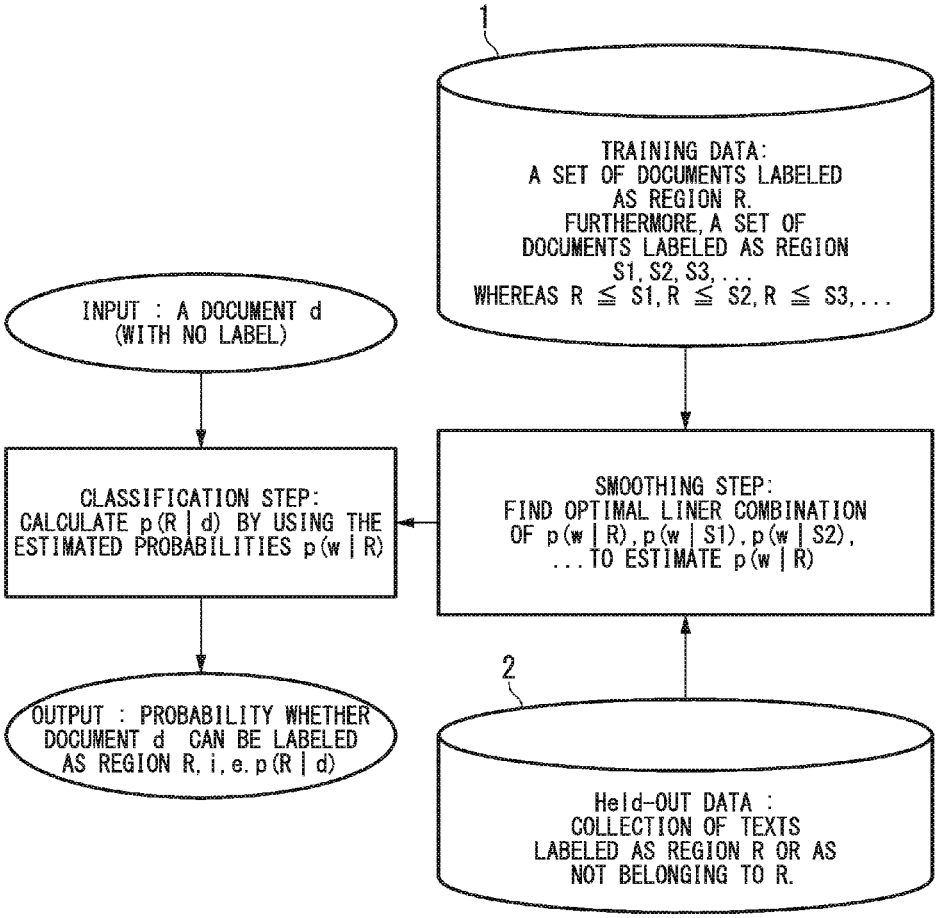

1
TRAINING DATA:
A SET OF DOCUMENTS LABELED
AS REGION R.
FURTHERMORE, A SET OF
DOCUMENTS LABELED AS REGION
S1, S2, S3, ...
WHEREAS R ≦ S1, R ≦ S2, R ≦ S3, ...
INPUT : A DOCUMENT d
(WITH NO LABEL)
CLASSIFICATION STEP:
CALCULATE p(R | d) BY USING THE
ESTIMATED PROBABILITIES p(w | R)
SMOOTHING STEP:
FIND OPTIMAL LINER COMBINATION
OF p(w | R), p(w | S1), p(w | S2),
...TO ESTIMATE p(w | R)
OUTPUT : PROBABILITY WHETHER
DOCUMENT d CAN BE LABELED
AS REGION R, i.e. p(R | d)
2
Held-OUT DATA :
COLLECTION OF TEXTS
LABELED AS REGION R OR AS
NOT BELONGING TO R.

FIG. 2

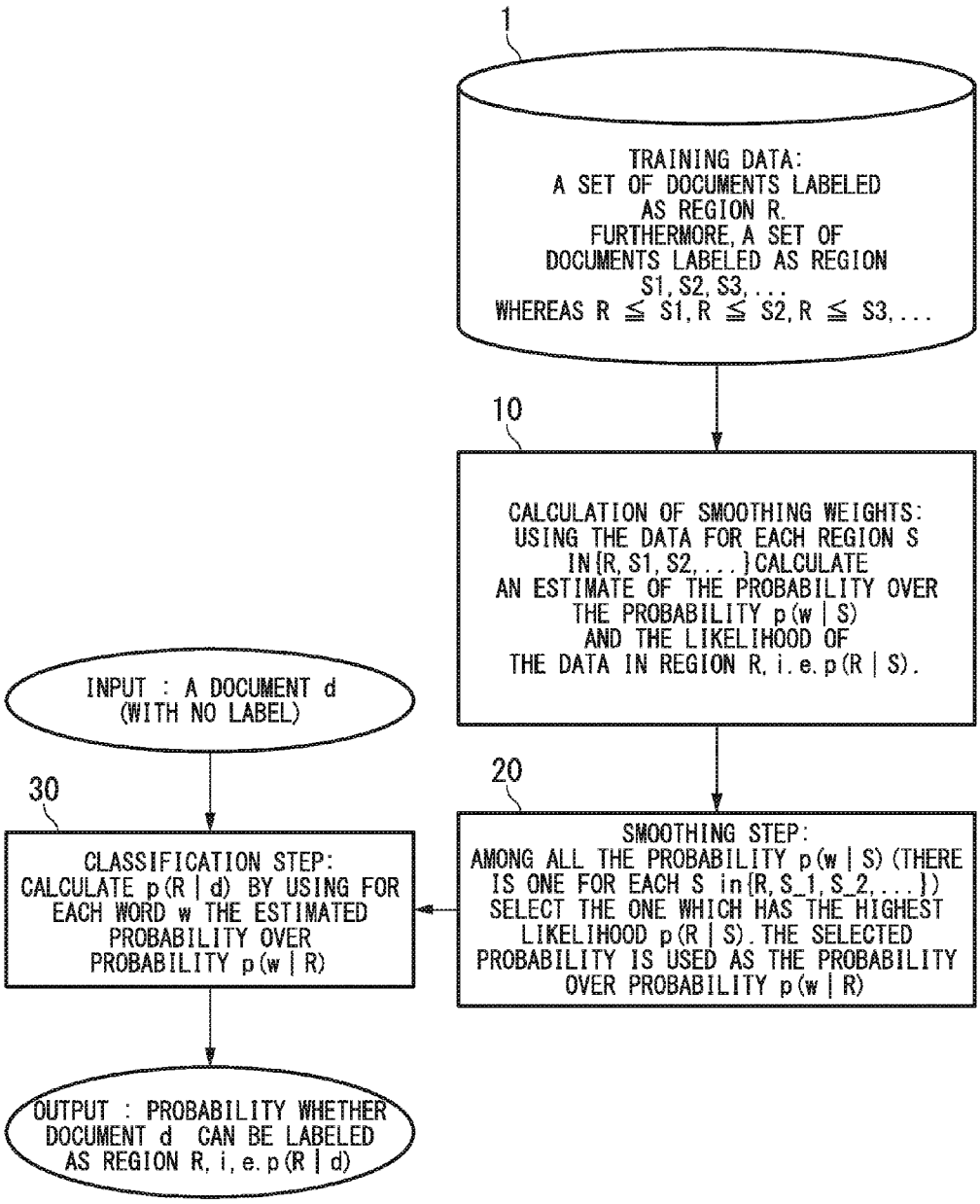

1
TRAINING DATA:
A SET OF DOCUMENTS LABELED AS REGION R.
FURTHERMORE, A SET OF DOCUMENTS LABELED AS REGION S1, S2, S3, ...
WHEREAS R ≦ S1, R ≦ S2, R ≦ S3, ...
10
CALCULATION OF SMOOTHING WEIGHTS:
USING THE DATA FOR EACH REGION S IN {R, S1, S2, ...} CALCULATE AN ESTIMATE OF THE PROBABILITY OVER THE PROBABILITY p(w | S) AND THE LIKELIHOOD OF THE DATA IN REGION R, i.e. p(R | S).
INPUT : A DOCUMENT d (WITH NO LABEL)
20
SMOOTHING STEP:
AMONG ALL THE PROBABILITY p(w | S) (THERE IS ONE FOR EACH S in {R, S_1, S_2, ...}) SELECT THE ONE WHICH HAS THE HIGHEST LIKELIHOOD p(R | S). THE SELECTED PROBABILITY IS USED AS THE PROBABILITY OVER PROBABILITY p(w | R)
30
CLASSIFICATION STEP:
CALCULATE p(R | d) BY USING FOR EACH WORD w THE ESTIMATED PROBABILITY OVER PROBABILITY p(w | R)
OUTPUT : PROBABILITY WHETHER DOCUMENT d CAN BE LABELED AS REGION R, i, e. p(R | d)

FIG. 3

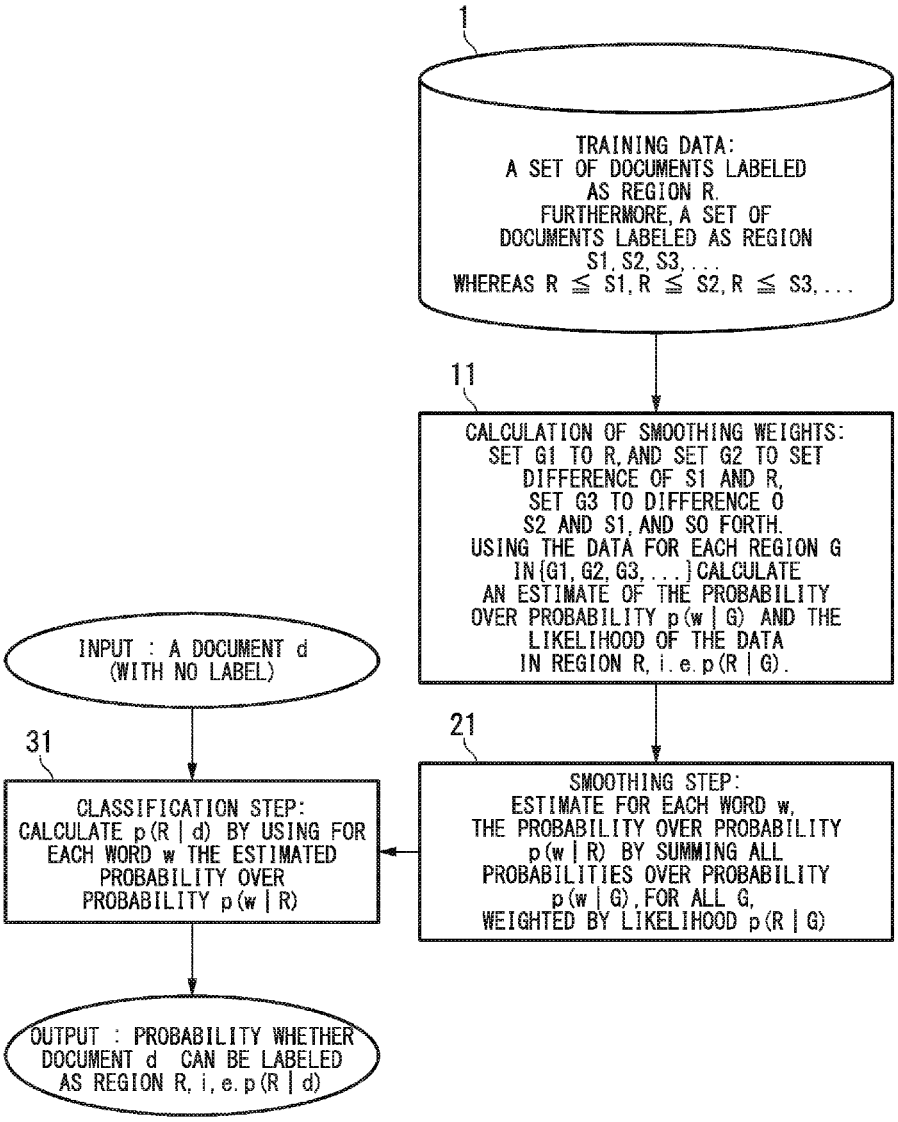

1
TRAINING DATA:
A SET OF DOCUMENTS LABELED AS REGION R.
FURTHERMORE, A SET OF DOCUMENTS LABELED AS REGION S1, S2, S3, ...
WHEREAS R ≦ S1, R ≦ S2, R ≦ S3, ...
11
CALCULATION OF SMOOTHING WEIGHTS:
SET G1 TO R, AND SET G2 TO SET DIFFERENCE OF S1 AND R,
SET G3 TO DIFFERENCE O S2 AND S1, AND SO FORTH.
USING THE DATA FOR EACH REGION G IN{G1, G2, G3, ...}CALCULATE AN ESTIMATE OF THE PROBABILITY OVER PROBABILITY p(w | G) AND THE LIKELIHOOD OF THE DATA IN REGION R, i.e. p(R | G).
21
SMOOTHING STEP:
ESTIMATE FOR EACH WORD w, THE PROBABILITY OVER PROBABILITY p(w | R) BY SUMMING ALL PROBABILITIES OVER PROBABILITY p(w | G), FOR ALL G, WEIGHTED BY LIKELIHOOD p(R | G)
31
CLASSIFICATION STEP:
CALCULATE p(R | d) BY USING FOR EACH WORD w THE ESTIMATED PROBABILITY OVER PROBABILITY p(w | R)
INPUT : A DOCUMENT d (WITH NO LABEL)
OUTPUT : PROBABILITY WHETHER DOCUMENT d CAN BE LABELED AS REGION R, i.e. p(R | d)

FIG. 4

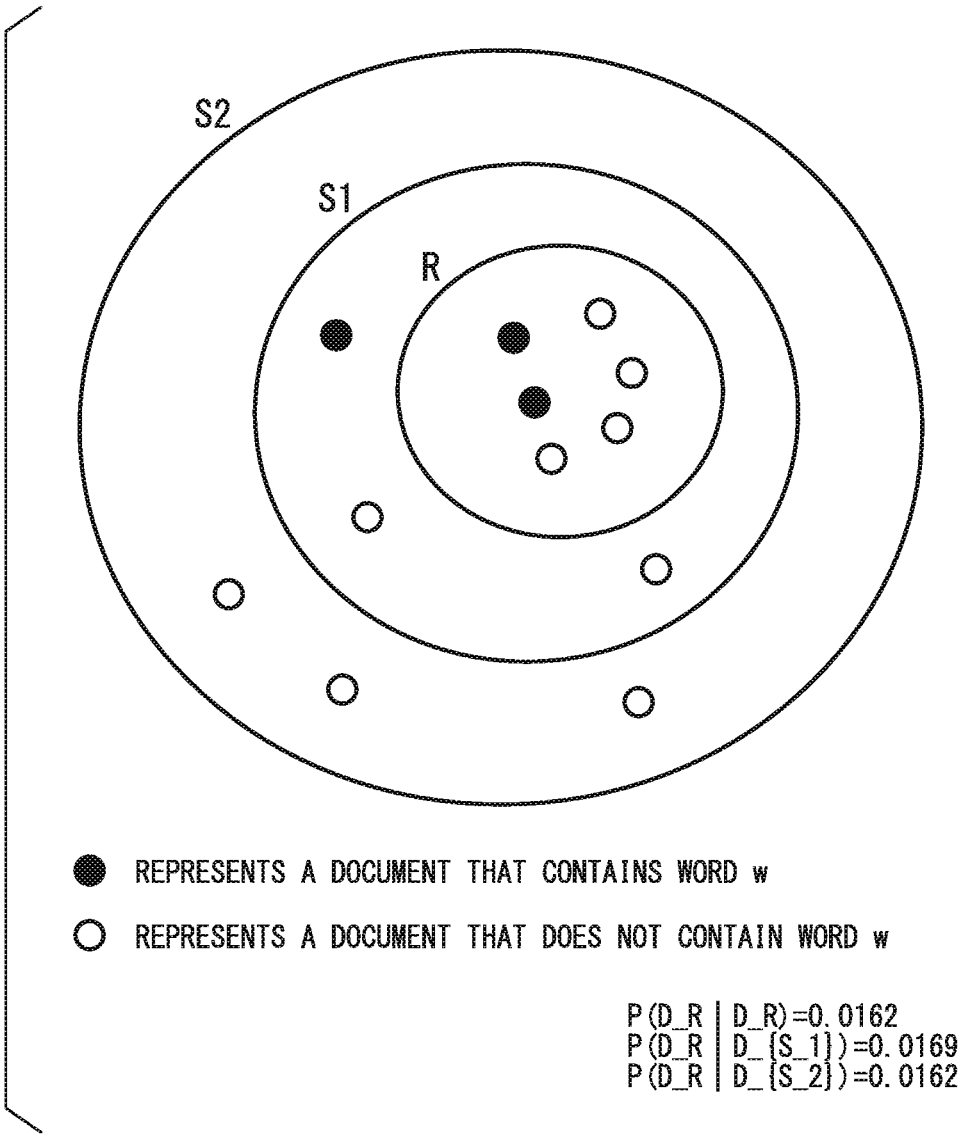
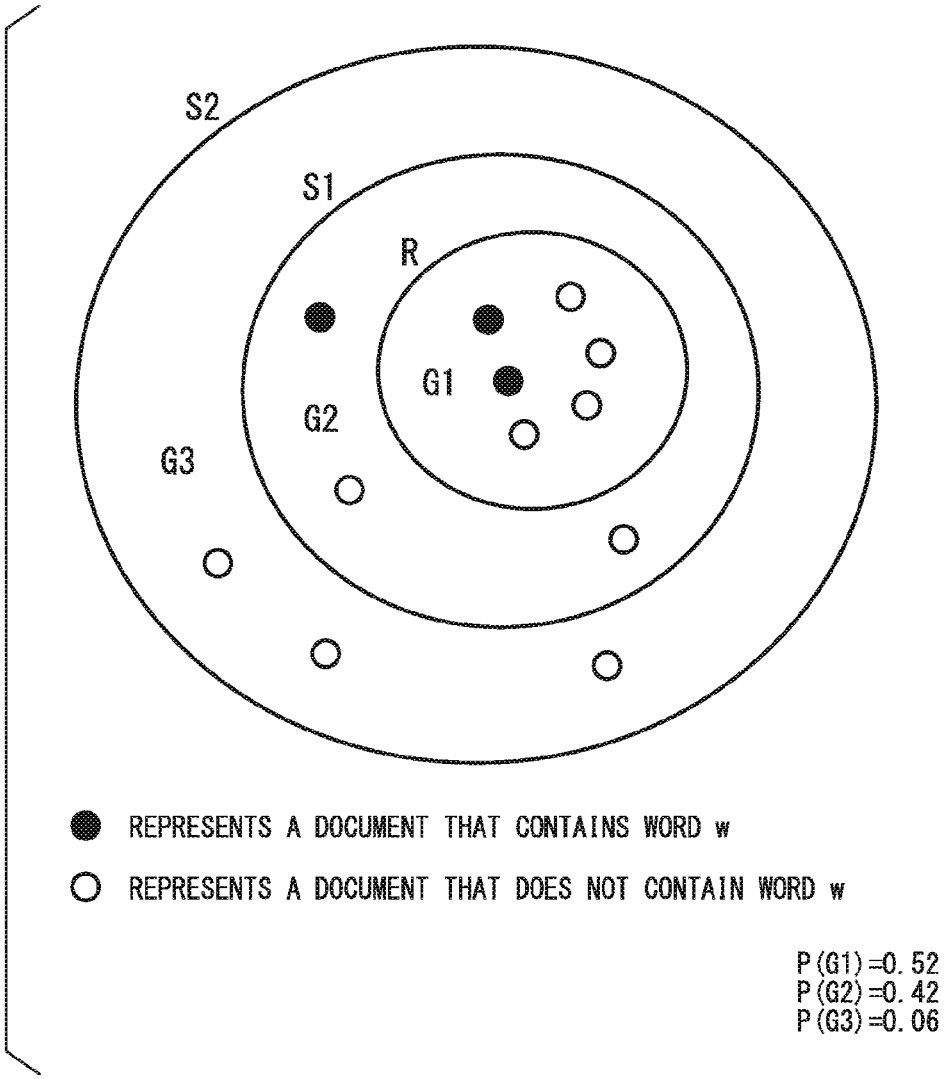

S2
S1
R
REPRESENTS A DOCUMENT THAT CONTAINS WORD w
REPRESENTS A DOCUMENT THAT DOES NOT CONTAIN WORD w
P(D_R | D_R)=0.0162
P(D_R | D_{S_1})=0.0169
P(D_R | D_{S_2})=0.0162

FIG. 5


S2
S1
R
G1
G2
G3
REPRESENTS A DOCUMENT THAT CONTAINS WORD w
REPRESENTS A DOCUMENT THAT DOES NOT CONTAIN WORD w
P(G1)=0.52
P(G2)=0.42
P(G3)=0.06

DOCUMENT CLASSIFICATION METHOD

TECHNICAL FIELD

[0001] The present invention relates to a method to decide whether a text document belongs to a certain class R or not (i.e. any other class), where there are only few training documents available for class R, and all classes can be arranged in a hierarchy.

BACKGROUND ART

[0002] The inventors of the present invention propose a smoothing technique that improves the classification of a text into two classes R and ¬ R, whereas only a few training instances for class R are available. The class ¬ R denote all classes that are class R, where all classes are arranged in a hierarchy. We assume that we have access to training instances of several classes that subsume class R.

[0003] This kind of problem occurs, for example, when we want to identify whether a document is about region (class) R, or not. For example, region R contains all geo-located Tweets (refer to messages from www.twitter.com) that belong to a certain city R, and outer regions S_1 and S_2 refer to the state, and the country, respectively, where city R is located. It is obvious that the classes R, S_1 and S_2 can be thought of being arranged in a hierarchy, where S_1 subsumes R, and S_2 subsumes S_1. However, most Tweets do not contain geo-location, i.e., we do not know whether the text messages were about region R. Given a small set of training data, we want to detect whether the text was about city R or not. In general, we have only a few training data instances available for city R, but much training data instances available for region S_1 and S_2.

[0004] Non-Patent Document 1 proposes for this task to use a kind of Naive Bayes classifier to decide whether a Tweet (document) belongs to region R. This classifier uses the word probabilities p(w|R) for classification (actually they estimate p(R|w), however, this difference is irrelevant here). In general R is small, and only a few training instance documents that belong to region R are available. Therefore, the word probabilities p(w|R) cannot be estimated reliable. In order to overcome this problem, they suggest to use training instance documents that belong to a region S that contains R.

[0005] Since S contains, in general, more training instances than R, Non-Patent Document 1 proposes to smooth the word probabilities p(w|R) by using p(w|S). For the smoothing they suggest to use a linear combination of p(w|R) and p(w|S), where the optimal parameter for the linear combination is estimated using held-out data.

[0006] This problem setting is also similar to hierarchical text classification. For example, class R is "Baseball in Japan", class S_1 is class "Baseball" and S_2 is class "Sports", and so forth. For this problem Non-Patent Document 2 suggests to smooth the word probabilities p(w|R) for class R by using one or more hyper-classes that contain class R. A hyper-class S has, in general, more training instances than class R, and therefore we can expect to get more reliable estimates. However, hyper-class S might also contain documents that are completely unrelated to class R. Non-Patent Document 2 relates to this dilemma as the trade-off between reliability and specificity. They solve this trade-off by setting weight λ that interpolates p(w|R) and p(w|S). The optimal weight λ needs to be set using held-out data.

Document of the Prior Art

[0007] Non-Patent Document 1: "You Are Where You Tweet: A Content-Based Approach to Geo-locating Twitter Users", Z. Cheng et. al., 2010.

[0008] Non-Patent Document 2: "Improving text classification by shrinkage in a hierarchy of classes", A. McCallum et al., 1998.

DISCLOSURE OF INVENTION

Problems to be Solved by the Invention

[0009] All previous methods require the use of held-out data 2 to estimate the degree of interpolation between p(w|R) and p(w|S), as shown in FIG. **1**. However, selecting a subset of training data instances of R (held-out data) reduces the data that can be used for training even further. This can out-weight the benefits that can be gained from setting the interpolation parameters with the held-out data. This problem is only partly mitigated by cross-validation, which, furthermore, can be computationally expensive. In FIG. **1**, X<=Y means document set Y contains document set X. Due to the analogy of geographic regions, we use the term "region", instead of the term "category" or "class".

[0010] It might appear that another obvious solution, would be to use the same training data twice, once for estimating the probability p(w|R) and once for estimating the optimal weight λ. However, using the approaches like described Non-Patent Document 1 or Non-Patent Document 2, would simply set the weight of λ to 1 for p(w|R) , and zero for p(w|S). This is because their method requires point-estimates of p(w|R) , which is a maximum-likelihood or maximum-a posterior estimate, that cannot measure the uncertainty of the estimate of p(w|R).

Means for Solving the Problem

[0011] Our approach compares the distributions of p(w|R) and p(w|S) and use the difference to decide if and how, the distribution p(w|R) should be smoothed using only the training data. The assumption of our approach can be summarized as follows: If the distribution of a word w is similar in region R and its outer region S, we expect that we can get a more reliable estimate of p(w|R) that is close to the true p(w|R) by using the sample space of region S. On the other hand, if the distributions are very different, we expect that we cannot do better than using the small sample size of R. The degree to which we can smooth the distribution p(w|R) with the distribution p(w|S) is determined by how likely it is that the training data instance of region R were generated by the distribution p(w|S). We denote this likelihood as $p(D_R|D_S)$. If, for example, we assume that the word occurrences are generated by a Bernoulli Trial, and we use as conjugate prior the Beta distribution, then the likelihood $p(D_R|D_S)$ can be calculated as the ratio of two Beta functions. In general, if the word occurrences are assumed to be generated by an i.i.d sample of distribution P with parameter vector θ, and conjugate prior f over the parameters θ, then the likelihood $p(D_R|D_S)$ can be calculated as a ratio of the normalization constants of two distributions of type f.

[0012] To make the uncertainty about the estimates p(w|R) (and p(w|S)) clear, we model the probability over these probabilities. For example, in case we assume that word occurrences are model by a Bernoulli distribution, we chose as the conjugate prior the beta distribution, and derive

therefore beta distribution for the probability over p(w|R) (and p(w|S)). For each probability over probability p(w|S) (there is one for each S∈{R, S_1, S_2, . . . }), we select the one which results in the highest likelihood of the data $p(D_R|D_S)$. We select this probability as the smoothed second-order word probability for p(w|R).

[0013] A variation of this approach is to first create mutual exclusive subsets R, G_1, G_2, . . . from the set {R, S_1, S_2, . . . }, and then calculate a weighted average of the probabilities over probability p(w|G), where the weights correspond to the data likelihood $p(D_R|D_G)$.

[0014] In the final step, for a new document d we calculate the probability that document d belongs to class R, by using the probability over probability p(w|R). For example, we use the naive Bayes assumption, and calculate p(d|R) by probability over probability p(w|R) (Bayesian Naive Bayes).

Effect of the Invention

[0015] The present invention has the effect of smoothing the probability that a word w occurs in a text that belongs to class R by using the word probabilities of outer-classes of R. It achieves this without the need to resort to additional held-out training data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] FIG. **1** is a block diagram showing the functional structure of the system proposed by previous work.

[0017] FIG. **2** is a block diagram showing a functional structure of a document clarification system according to a first exemplary embodiment of the present invention.

[0018] FIG. **3** is a block diagram showing a functional structure of a document clarification system according to a second exemplary embodiment of the present invention.

[0019] FIG. **4** shows an example related to the first embodiment.

[0020] FIG. **5** shows an example related to the second embodiment.

EXEMPLARY EMBODIMENTS FOR CARRYING OUT THE INVENTION

First Exemplary Embodiment

[0021] The main architecture usually performed by a computer system is described in FIG. **2**. We assume we are interested in whether the text is about region R or not, which we denote by ¬ R. Due to the analogy of geographic regions we use the term "region", but it is clear that this can be more abstractly considered as a "category" or "class". Further in FIG. **2**, X<=Y means document set Y contains document set X.

[0022] Let θ be a vector of parameters of our model that generates all training documents D stored in a non-transitory computer storage medium 1 such as a hard disk drive. Our approach tries to optimize the probability p(D) as follows:

$$p(D)=\int p(D|\theta)p(\theta)d\theta.$$

[0023] In the following, we will focus on p(D|θ) which can be calculated as follows:

$$p(D\mid\theta)=\sum_{i\in D}p(d_i,l(d_i)\mid\theta)=\prod_i p(d_i\mid l(d_i),\theta)\cdot p(l(d_i)\mid\theta)$$

where D is the training data which contains the documents $\{d_1, d_2, \ldots\}$, and the corresponding label for each document d_i is denote $l(d_i)$ (the first equality holds due to the i.i.d assumption). In our situation, $l(d_i)$ is either the label saying that the document d_i belongs to region R, or the label saying that it does not belong to region R, i.e., $l(d_i)\in\{R, \neg R\}$.

[0024] Our model uses the naive Bayes assumption and therefore it holds:

$$\prod_i p(d_i\mid l(d_i),\theta)\cdot p(l(d_i)\mid\theta)=\prod_i p(l(d_i)\mid\theta)\cdot\prod_{w\in F}p(w\mid l(d_i),\theta)$$
$$=\left(\prod_i p(l(d_i)\mid\theta)\right)\cdot\left(\prod_i\prod_{w\in F}p(w\mid l(d_i),\theta)\right)$$

[0025] The set of words F is our feature space. It can contain all words that occurred in the training data D, or a subset (e.g., only named entities). Our model assumes that, given a document that belongs to region R, a word w is generated by a Bernoulli distribution with probability θ_w. Analogously, for a document that belongs to region ¬ R, word w is generated by a Bernoulli distribution with probability ϑ_w. That means, we distinguish here only the two cases, that is whether a word w occurs (one or more times) in a document, or whether it does not occur.

[0026] We assume that we can reliably estimate $p(l(d_i)|\theta)$ using a maximum likelihood approach, and therefore focus on the term $\Pi_i\Pi_{w\in F}p(w|l(d_i),\theta)$.

$$\prod_{i\in D}\prod_{w\in F}p(w\mid l(d_i),\theta)=\prod_{w\in F}\theta_w^{c_w}\cdot(1-\theta_w)^{n_R-c_w}\cdot\vartheta_w^{d_w}\cdot(1-\vartheta_w)^{n_{\neg R}-d_w},$$

where n_R and $n_{\neg R}$ is the number of documents that belong to R, and ¬ R, respectively; c_w, is the number of documents that belong to R and contain word w, analogously d_w is the number of documents that belong to ¬ R and contain word w. Since we assume that the region ¬ R is very large, that is $n_{\neg R}$ is very large, we can use a maximum likelihood (or maximum a-posterior with low informative prior) estimate for θ. Therefore, our focus, is on how to estimate θ_w), or more precisely speaking, how to estimate the distribution $p(\theta_w)$.

Our choice of one θ_w, will affect p(D|θ) only by the factor:

$$\theta_w^{c_w}\cdot(1-\theta_w)^{n_R-c_w}. \quad (1)$$

[0027] This factor actually corresponds to the probability $p(D_R|\theta_w)$, where D_R is the set of (training) documents that belong to region R.

[Estimating $p(\theta_w)$]

[0028] First, recall that the probability θ_W corresponds to the probability p(w|R), i.e., the probability that a document that belongs to region R, contains the word w (one or more times). For estimating the probability $p(\theta_w)$ we use that the words were generated by a Bernoulli trial. The sample size of this Bernoulli trial is:

$$n_R:=|\{d|l(d)=R\}|$$

[0029] Using this model, we can derive the maximum likelihood estimate of p(w|R) which is:

$$ML(p(w \mid R))_R = \frac{c_R(w)}{n_R},$$

[0030] where we denote by $c_R(w)$ the number of documents in region R that contain word w. The problem with this estimate is, that it is unreliable if n_R is small. Therefore, we suggest to use as an estimate a region S which contains R and is larger than or equal to R, i.e., $n_S \geq n_R$. The maximum likelihood estimate of p(w|R) becomes:

$$ML(p(w \mid R))_S = \frac{c_S(w)}{n_S}.$$

[0031] This way, we can get a more robust estimate of the true (but unknown) probability p(w|R). However, it is obvious that it biased towards the probability of p(w|S). If we knew that the true probabilities of p(w|S) and p(w|R) are identical, then the estimate $ML(p(w|R))_S$ will give us a better estimate than $ML(p(w|R))_R$. Obviously, there is a trade off when choosing S: if S is almost the same size as R, then there is a high chance that the true probability of p(w|S) and p(w|R) are identical.

[0032] However the same sample size hardly increases. On the other hand, if S is very large, there is a high chance that the true probability of p(w|S) and p(w|R) are different. This trade-off is sometimes also referred as the trade-off between specificity and reliability (see Non-Patent Document 2). Let D_R denote the observed documents in region R. The obvious solution to estimate $p(\theta_w)$ is to use $p(\theta_w|D_R)$ which is calculated by:

$$p(\theta_w|D_R) \propto p(D_R|\theta_w) \cdot p_0(\theta_w)$$

where for the prior $p_0(\theta_w)$ we use a beta-distribution with hyper-parameters α_0 and β_0. We can now write:

$$p(\theta_w|D_R) \propto \theta_w^{c_R} \cdot (1-\theta_w)^{n_R - c_R} \cdot \theta_w^{\alpha_0 - 1} (1-\theta_w)^{\beta_0 - 1},$$

where we wrote c_R short for $c_R(w)$. (Also in the following, if it is clear from the context that we refer to word w, we will simply write c_R instead of $c_R(w)$.)

[0033] However, in our situation the sample size n_R is small, which will result in a relatively flat, i.e., low informative distribution of θ_w. Therefore, our approach suggests to use S with its larger sample size n_S to estimate a probability distribution over θ_w. Let D_S denote the observed documents in region S. We estimate $p(\theta_w)$ with $p(\theta_w - D_S)$ which is calculated, analogously to $p(\theta_w|D_S)$, by:

$$p(\theta_w|D_S) \propto \theta_w^{c_S} \cdot (1-\theta_w)^{n_S - c_S} \cdot \theta_w^{\alpha_0 - 1} (1-\theta_w)^{\beta_0 - 1}.$$

[0034] Making the normalization factor explicit this can be written as:

$$p(\theta_w \mid D_S) = \frac{1}{B(c_S + \alpha_0,\, n_S - c_S + \beta_0)} \cdot \theta_w^{c_S + \alpha_0 - 1} \cdot (1-\theta_w)^{n_S - c_S + \beta_0 - 1}, \quad (2)$$

where $B(\alpha, \beta)$ is the Beta function.

[0035] Our goal is to find the optimal S, whereas we define optimal as the $S \geq R$ that maximizes the probability of the observed data (training data) D, i.e., p(D). Since we focus on the estimation of the occurrence probability in region R (i.e., θ_w), it is sufficient to maximize $p(D_R)$ (this is because $p(D) = p(D_R) \cdot p(D_{\neg R})$, and $p(D_{\neg R})$ is constant with respect to θ_w). $p(D_R)$ can be calculated as follows:

$$p(D_R) = \prod_w E_{p(\theta)}[p(D_R \mid \theta_w, D_S)] = \prod_w p_w(D_R),$$

where we define $E_{p(\Theta)}[p(D_R|\theta_w, D_S)]$ as $p_w(D_R)$. In order to make it explicitly clear that we use D_S to estimate the probability $p(\theta_w)$, we write $p_w(D_R|D_S)$, instead of $p_w(D_R)$. $p_w(D_R|D_S)$ is calculated as follows:

$$p_w(D_R|D_S) = E_{p(\Theta)}[p(D_R|\theta_w, D_S)] = \int p(D_R|\theta_w, D_S) p(\theta_w|D_S) d\theta_w = \int p(D_R|\theta_w) p(\theta_w|D_S) d\theta_w$$

[0036] Using Equation (1) and Equation (2) we can write:

$$p_w(D_R \mid D_S) = \frac{1}{B(c_S + \alpha_0,\, n_S - c_S + \beta_0)} \cdot \int \theta_w^{c_S + \alpha_0 - 1} \cdot (1-\theta_w)^{n_S - c_S + \beta_0 - 1} \cdot \theta_w^{c_w} \cdot (1-\theta_w)^{n_R - c_w} d\theta_w$$

[0037] Note that the latter term is just the normalization constant of a beta distribution since:

$$\int \theta_w^{c_S + \alpha_0 - 1} \cdot (1-\theta_w)^{n_S - c_S + \beta_0 - 1} \cdot \theta_w^{c_w} \cdot (1-\theta_w)^{n_R - c_w} d\theta_w = \int \theta_w^{c_S + \alpha_0 - 1 + c_w} \cdot (1-\theta_w)^{n_S - c_S + \beta_0 - 1 + n_R - c_w} = B(c_S + \alpha_0 + c_w,\, n_S - c_S + \beta_0 + n_R - c_w)$$

[0038] Therefore $p_w(D_R|D_S)$ can be simply calculated as follows:

$$p_w(D_R \mid D_S) = \frac{B(c_S + \alpha_0 + c_w,\, n_S - c_S + \beta_0 + n_R - c_w)}{B(c_S + \alpha_0,\, n_S - c_S + \beta_0)}. \quad (3)$$

[0039] We can summarize our procedure for estimating $p(\theta_w)$ as follows. Given several candidates for S, i.e., S_1, S_2, S_3, . . . , we select the optimal S* for estimating $p(\theta_w)$ by using:

$$S^* = \underset{S \in \{S_1, S_2, S_3, \ldots\}}{\operatorname{argmax}} \; p_w(D_R \mid D_S) \quad (4)$$

whereas $p(D_R|D_S)$ is calculated using Equation (3). Note that, in general, for each word w a different outer region S is optimal. The estimate for $p(\theta_w)$ is then:

$$p(\theta_w|D_{S^*}).$$

[0040] The calculation of $p(\theta_w|D_{S^*})$ can be considered as calculating a smoothed estimate for θ_w, this refers to component **10** in FIG. **2**; moreover choosing the optimal smoothed weight with respect to $p(D_R|D_S)$ is referred to as

component 20 in FIG. **2**. A variation of this approach is to use the same outer region S, for all w, whereas the optimal region S* is selected using:

$$S^* = \operatorname*{argmax}_{S\in\{S_1,S_2,S_3,\dots\}} p(D_R \mid D_S) = \operatorname*{argmax}_{S\in\{S_1,S_2,S_3,\dots\}} \prod_{w\in F} p_w(D_R \mid D_S). \quad (5)$$

[0041] An example is given in FIG. **4**.

[Classification]

[0042] We show here how to use the estimates $p(\theta_w)$, for each word w∈F, to decide for a new document d whether it belongs to region R or not. Note that document d is not in training data D. This corresponds to component **30** in FIG. **2** and component **31** in FIG. **3**. For this classification, we use the training data D with the model, which we described above as follows:

$$\operatorname*{argmax}_{l\in R,\neg R} p(l(d) = l \mid D, d)$$

[0043] The probability can be calculated as follows:

$$p(l(d)=l|D,d)\propto p(d|D,l(d)=l)\cdot p(l(d)=l|D)$$

We assume that D is sufficiently large and therefore estimate p(l(d)=l|D) with maximum-likelihood (ML) or maximum-a posterior (MAP) approach. p(d|D, l(d)=l) is calculated as follows:

$$p(d|D,l(d)=l)=\int p(d|\theta,\theta,D,l(d)=l)\cdot p(\theta,\theta|D,l(d)=l)\,d\theta d\theta$$

Where θ and θ are each vector of parameters that contains for each word w the probability θ_w, and θ_w, respectively. For l=¬ R we can simply use the ML or MAP for estimate for θ estimate since we assume that $D_{\neg R}$ is sufficiently large. For the case l=R we have:

$$p(d \mid D, l(d) = l) = \int p(d \mid \theta, D, l(d) = l)\cdot p(\theta \mid D, l(d) = l)d\theta$$
$$= \int \prod_{w\in F} \theta_w^{d_w}\cdot(1-\theta_w)^{d_w}\cdot p(\theta_w \mid S_w^*)d\theta,$$

where S^*_w is the optimal S for a word w that we specified in Equation (4), or we set S^*_w independent of w to the value specified in Equation (5); d_w is defined to be 1, if w∈d, otherwise 0.

[0044] Integrating over all possible choices of θ_w for calculating p(d|D, l(d)=l) is sometimes referred to as Bayesian Naive Bayes (see, for example, "Bayesian Reasoning and Machine Learning", D. Barber, 2010, pages 208-210). We note that instead of integrating over all possible values for θ_w, we can use a point-estimate of θ_w, like for example the following (smoothed) ML-estimated:

$$\theta_w := ML(p(w \mid R))_{S^*} = \frac{c^*_S(w)}{n_{S^*}}.$$

Second Exemplary Embodiment

[0045] Instead of selecting only one S for estimating $p(\theta_w)$, we can use region R and all its available outer-regions S_1, S_2, . . . and weight them appropriately. This idea is outlined in FIG. **3**. First, assume that we are given regions G_1, G_2, . . . that are mutually exclusive. As before, our estimate for $p(\theta_w)$ is $p(\theta_w|D_{Gi})$, if we assume that G_i is the best region to use to estimate θ_w. The calculation of G_i and $p(\theta_w|D_{Gi})$ is referred to as component **11** in FIG. **3**. However, in contrast to before, instead of choosing only one G_i, we select all and weight them by the probability that G_i is the best region to estimate θ_w. We denote this probability $p(G_i)$. Then, the estimate for θ_w can be written as:

$$p(\theta_w) = \sum_{G\in\{G_1,G_2,\dots\}} p(\theta_w \mid D_G)\cdot p(D_G) \quad (200)$$

We assume that:

$$\sum_{G\in\{G_1,G_2,\dots\}} p(D_G) = 1,$$

and

$$p(D_G) \propto p(D_R \mid D_G),$$

where the probability $p(D_R|D_G)$ is calculated as described in Equation (3). In words, this means, we assume that the probability that G is the best region to estimate $p(\theta_w)$ is proportional to the likelihood $p(D_R|D_G)$. Recall that $p(D_R|D_G)$ is the likelihood that we observer the training data D_R when we estimate $p(\theta_w)$ with D_G. The calculation of $p(\theta_w)$ using Equation (200) is referred to component **21** in FIG. **3**.

[0046] In our setting, we have that S_1, S_2, . . . are all outer-regions of R, and thus, not mutually exclusive. Therefore we define the regions G_1, G_2, . . . as follows:

$$G_1 := R,\ G_2 := S_1 \setminus R,\ G_3 := S_2 \setminus S_1,\ G_4 := S_3 \setminus S_2, \dots$$

where we assume that $R \subset S_1 \subset S_2 \subset S_3 \dots$.

[0047] An example is given in FIG. **5** which shows the same (training) data as in FIG. **4** together with the corresponding mutual exclusive regions G_1, G_2 and G_3. G_1 is identical to R which contains 6 documents, out of which 2 documents contain the word w. G_2 contains 3 documents, out of which 1 document contains the word w. G_3 contains 3 documents, out of which no document contains the word w. Using Equation (3) we get:

$$p(D_R|D_{G_1})=0.0153$$

$$p(D_R|D_{G_2})=0.0123$$

$$p(D_R|D_{G_3})=0.0017$$

And since the probabilities for $p(D_G)$ must sum to 1, we get:

$$p(D_{G_1})=0.52$$

$$p(D_{G_2})=0.42$$

$$p(D_{G_3})=0.06$$

[0048] The document classification method of the above exemplary embodiments may be realized by dedicated hard-

ware, or may be configured by means of memory and a DSP (digital signal processor) or other computation and processing device. On the other hand, the functions may be realized by execution of a program used to realize the steps of the document classification method.

[0049] Moreover, a program to realize the steps of the document classification method may be recorded on computer-readable storage media, and the program recorded on this storage media may be read and executed by a computer system to perform document classification processing. Here, a "computer system" may include an OS, peripheral equipment, or other hardware.

[0050] Further, "computer-readable storage media" means a flexible disk, magneto-optical disc, ROM, flash memory or other writable nonvolatile memory, CD-ROM or other removable media, or a hard disk or other storage system incorporated within a computer system.

[0051] Further, "computer readable storage media" also includes members which hold the program for a fixed length of time, such as volatile memory (for example, DRAM (dynamic random access memory)) within a computer system serving as a server or client, when the program is transmitted via the Internet, other networks, telephone circuits, or other communication circuits.

INDUSTRIAL APPLICABILITY

[0052] The present invention allows to accurately estimate whether a tweet is about a small region R or not. A tweet might report about a critical event like an earthquake, but not knowing from which region the tweet was sent, renders the information useless. Unfortunately, most Tweets do not contain geolocation information which makes it necessary to estimate the location based on the text content. The text can contain words that mention regional shops or regional dialects which can help to decide whether the Tweet was sent from a certain region R or not. It is clear that we would like keep the classification results accurate, if region R becomes small. However, as R becomes small only a fraction of training data instances become available to estimate whether the tweet is about region R or not.

[0053] Another important application is to decide whether a text is about a certain predefined class R, or not, where R is a sub-class of one or more other classes. This problem setting is typical in hierarchical text classification. For example, we would like to know whether the text belongs to class "Baseball in Japan", whereas this class is a sub-class of "Baseball" that in turn is a sub-class of "Sports", and so forth.

1. A document classification method comprising:

a first step for calculating smoothing weights for each word w and a fixed class R, the first step including, given a set of classes $\{R, S_1, S_2, \ldots\}$ where class R is subsumed by class S_1, class S_1 is subsumed by class S_2, . . . , calculating for each class S probability over probability p(w|S) representing probability that word w occurs in a document belonging to class S, and, for each of these probabilities over the probabilities p(w|S), calculating the likelihood of the training data observed in class R;

a second step for calculating smoothed second-order word probability, the second step including, among all the probabilities over the probability p(w|S) (there is one for each $S \in \{R, S1, S2, \ldots\}$), selecting the one which results in the highest likelihood of the data as calculated in the second step before, the selected probability being used as the smoothed second-order word probability for p(w|R); and

a third step for classifying document including calculating the probability that the document belongs to the class R by using the smoothed second-order word probability to integrate over all possible choices of p(w|R), or by using the maximum a-posteriori estimate of the smoothed estimated of p(w|R).

2. The document classification method according to claim **1**, wherein the first step further includes denoting R as G_1, denoting set differences of the documents in R and S_1 as G_2, denoting set difference of the documents in S_1 and S_2 as G_3, . . . , for each G in $\{G_1, G_2, G_3, \ldots\}$, calculating the probability over the probability p(w|G) representing probability that word w occurs in a document belonging to document set G, and for each of these probabilities over the probabilities p(w|G), calculating the likelihood of the training data observed in class R; and

the second step further includes calculating smoothed second-order word probabilities including calculating the probability over the word probability p(w|R) by using the weighted sum of the probabilities of the probability p(w|G) calculated in the step before, where the weights correspond to the likelihoods calculated in the step before.

* * * * *