Probabilistic Semantic Similarity Measurements for Noisy Short Texts Using Wikipedia Entities

Masumi Shirakawa[1], Kotaro Nakayama[2], Takahiro Hara[1], Shojiro Nishio[1]

[1]Osaka University, Osaka, Japan

[2]University of Tokyo, Tokyo, Japan

Challenge in short text analysis

Statistics are not always enough.

A year and a half after Google pulled its popular search engine out of mainland China

Baidu and Microsoft did not disclose terms of the agreement

Challenge in short text analysis

Statistics are not always enough.

A year and a half after Google pulled its popular search engine out of mainland China

Baidu and Microsoft did not disclose terms of the agreement

They are talking about...

Search engines and China

Challenge in short text analysis

Statistics are not always enough.

A year and a half after Google pulled its popular search engine out of mainland China

Baidu and Microsoft did not disclose terms of the agreement

They are talking about...

Search engines and China

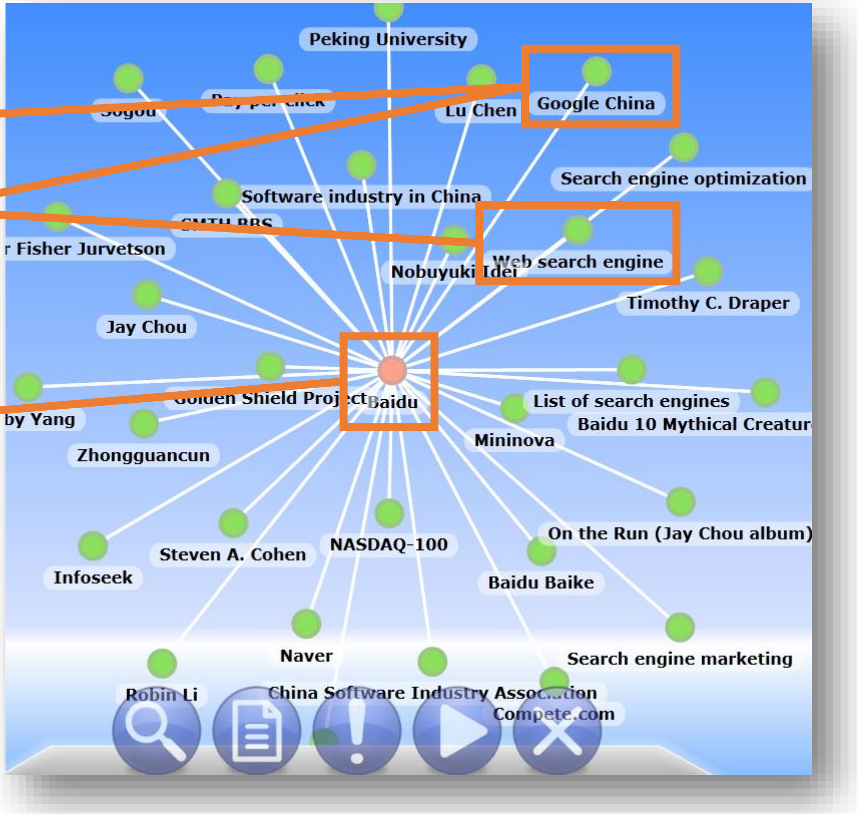
How do machines know that the two sentences mention about the similar topic?

Reasonable solution

Use external knowledge.

A year and a half after Google pulled its popular search engine out of mainland China

Baidu and Microsoft did not disclose terms of the agreement



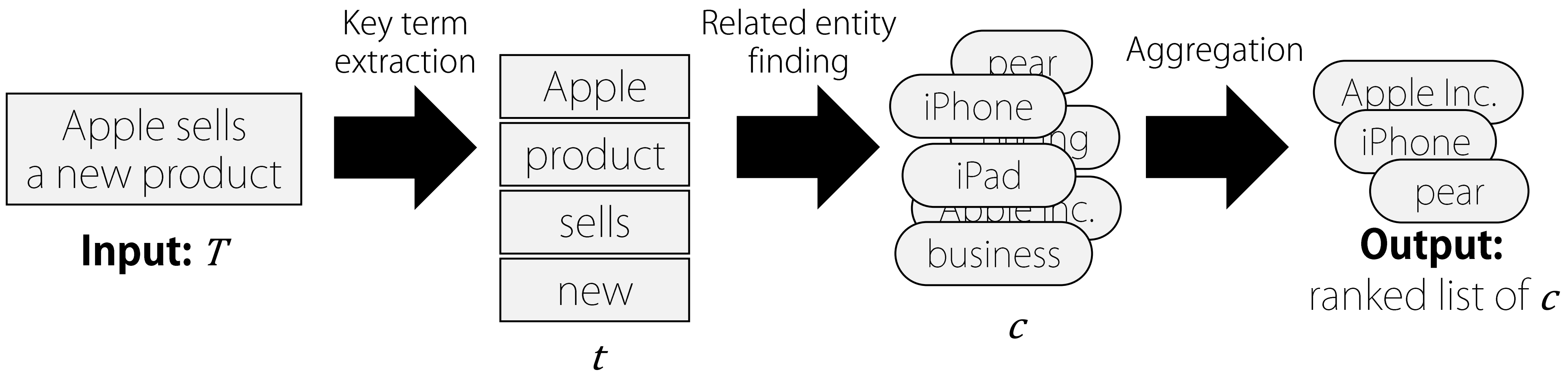
Wikipedia Thesaurus [Nakayama06]

Related work

ESA: Explicit Semantic Analysis [Gabrilovich07]

Add Wikipedia articles (entities) to a text as its semantic representation.

1. Get search ranking of Wikipedia for each term (i.e. Wiki articles and scores).
2. Simply sum up the scores for aggregation.



Problems in real world noisy short texts

"Noisy" means semantically noisy in this work.
(We do not handle informal or casual surface forms, or misspells)

Term ambiguity

- Apple (fruit) should not be related with Microsoft.

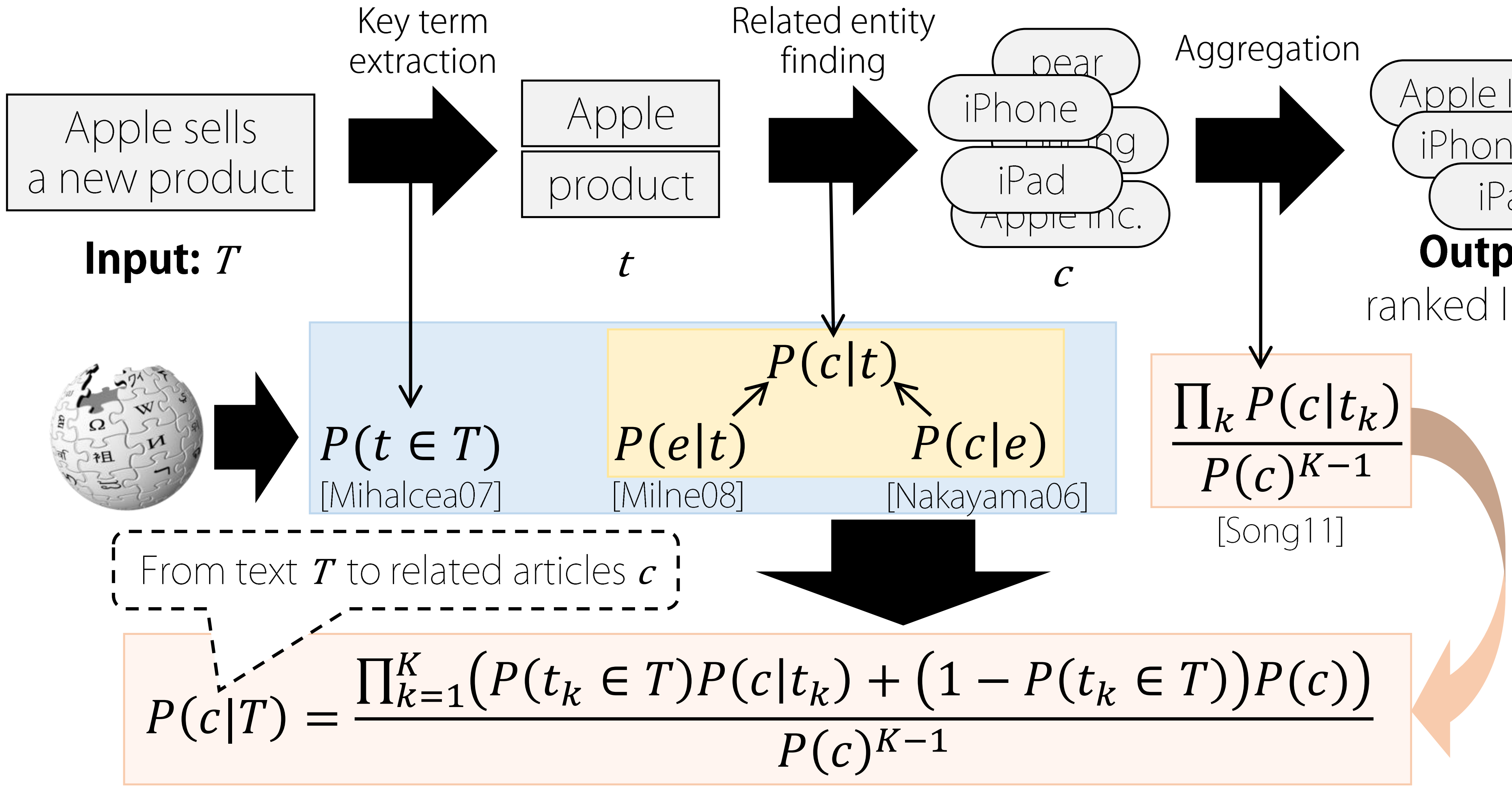
Fluctuation of term dominance

- A term is not always important in texts.

We explore more effective aggregation method.

Probabilistic method

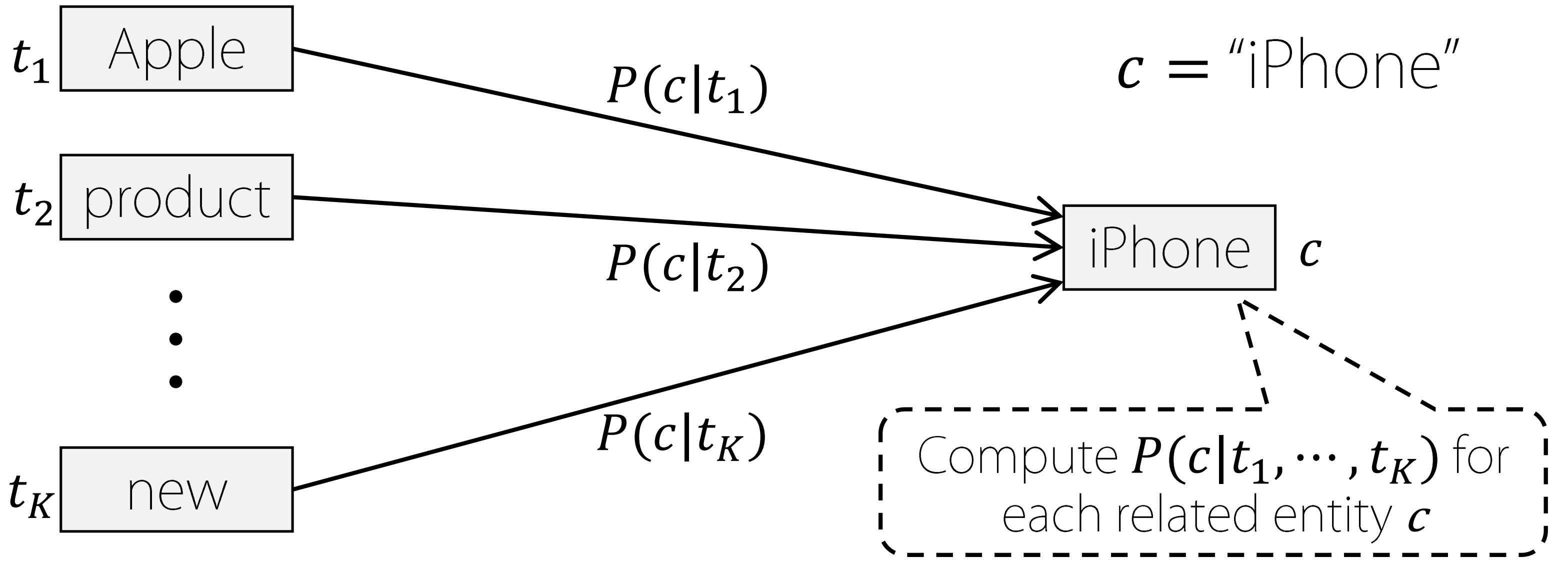
We propose **Extended naïve Bayes** to aggregate related entities



When input is multiple terms

Apply naïve Bayes [Song11] to multiple terms $t_1, \dots, t_K$
to obtain related entity c using each probability $P(c|t_k)$.

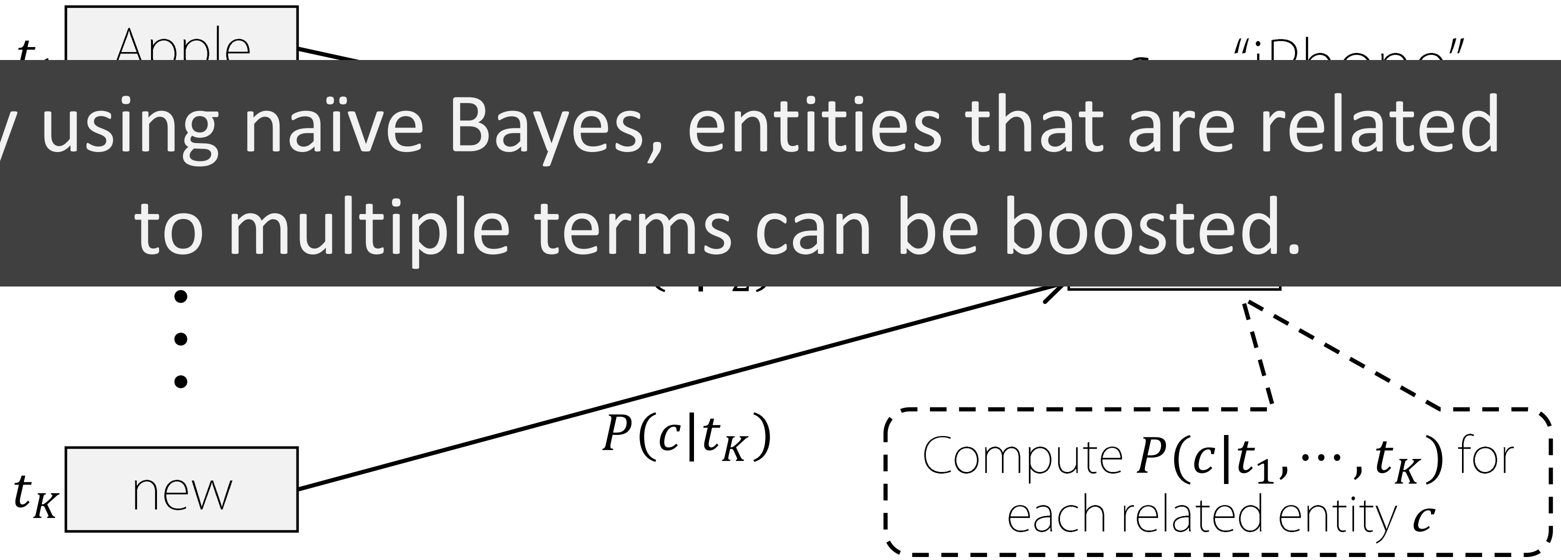
$$P(c|t_1, \dots, t_K) = \frac{P(t_1, \dots, t_K|c)P(c)}{P(t_1, \dots, t_K)} = \frac{P(c)\prod_k P(t_k|c)}{P(t_1, \dots, t_K)} = \frac{\prod_k P(c|t_k)}{P(c)^{K-1}}$$



When input is multiple terms

Apply naïve Bayes [Song11] to multiple terms $t_1, \dots, t_K$
to obtain related entity c using each probability $P(c|t_k)$.

$$P(c|t_1,\dots,t_K) = \frac{P(t_1,\dots,t_K|c)P(c)}{P(t_1,\dots,t_K)} = \frac{P(c)\prod_k P(t_k|c)}{P(t_1,\dots,t_K)} = \frac{\prod_k P(c|t_k)}{P(c)^{K-1}}$$

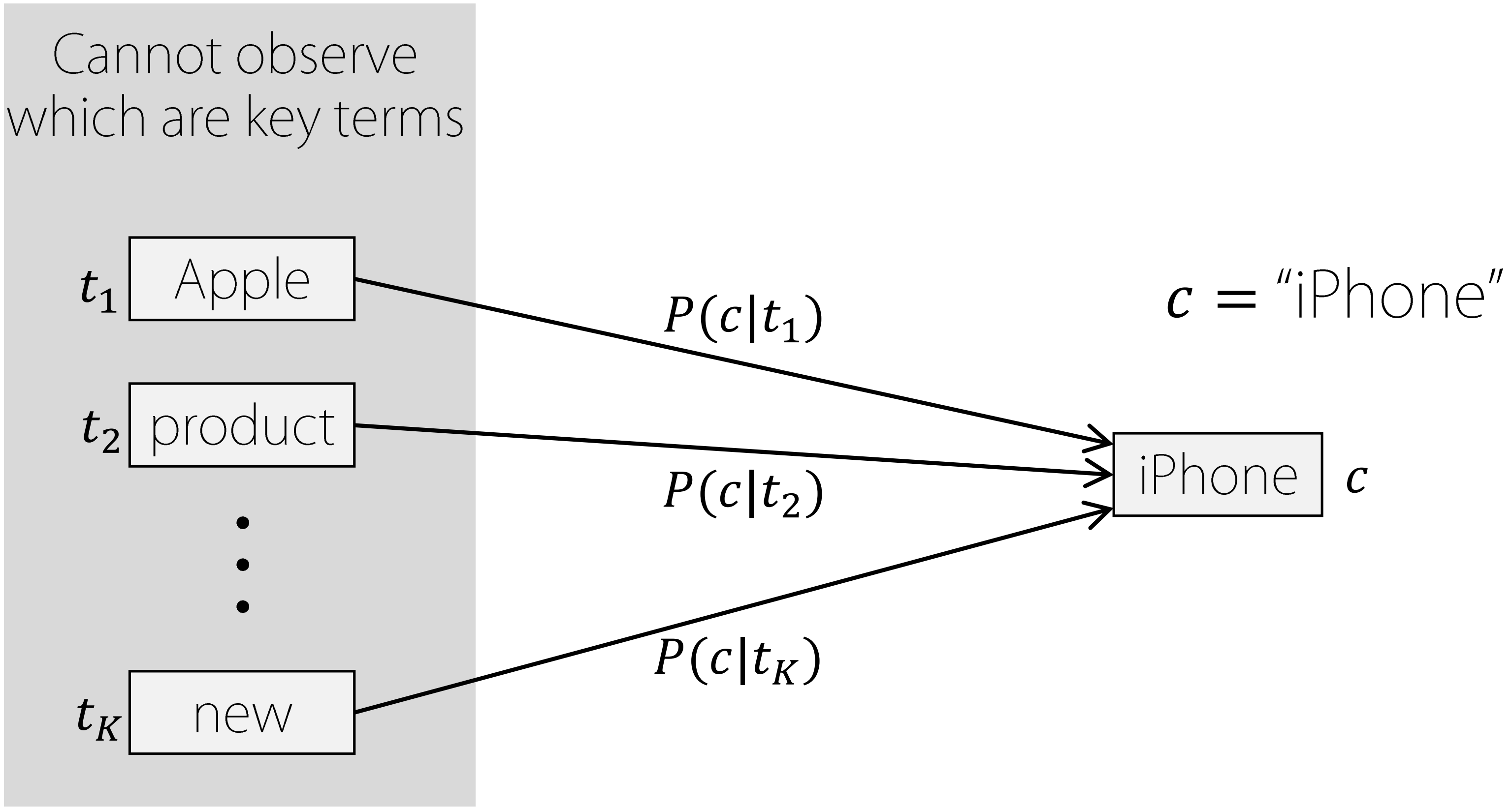


By using naïve Bayes, entities that are related to multiple terms can be boosted.

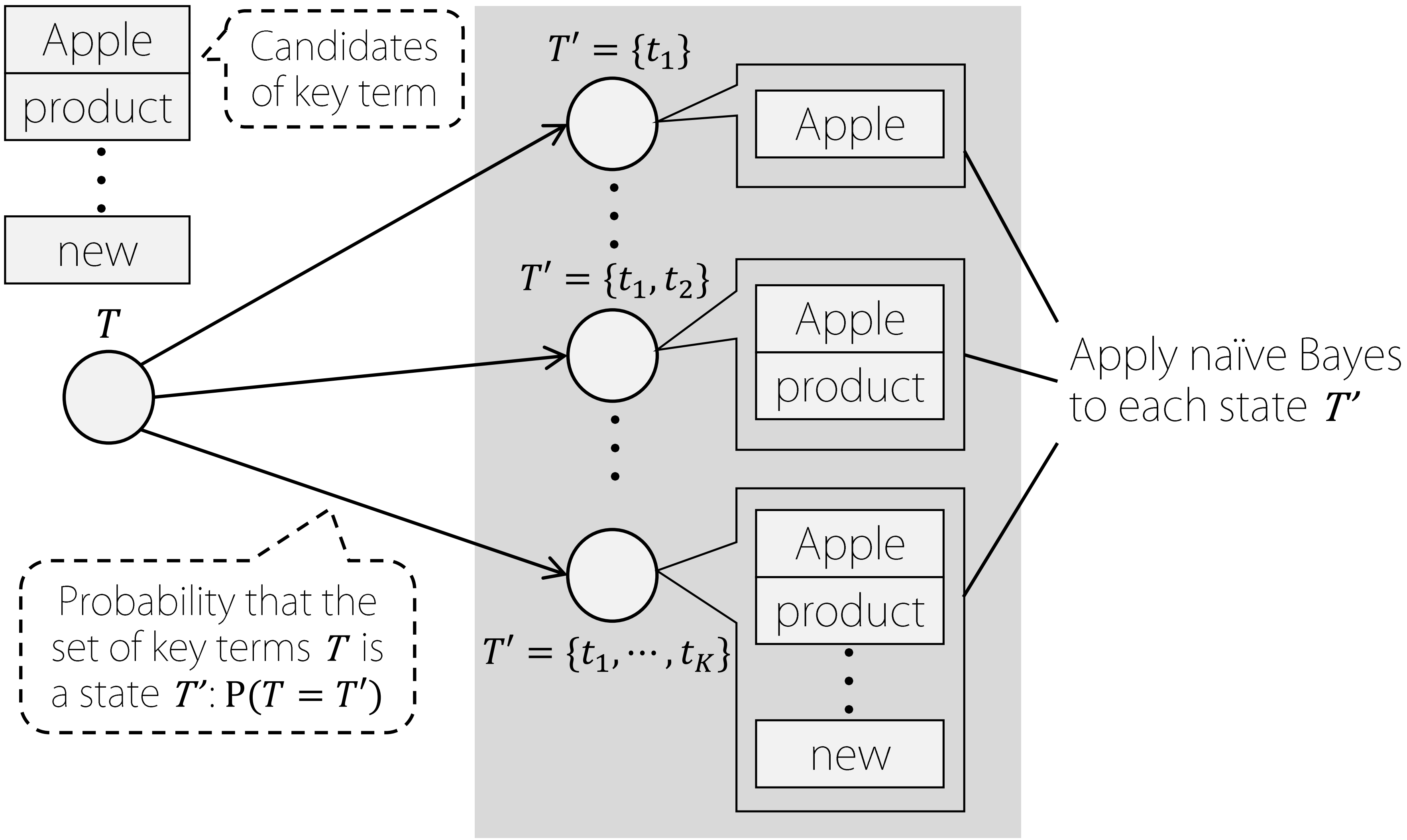
When input is text

Not "multiple terms" but "text," i.e., we don't know which terms are key terms.

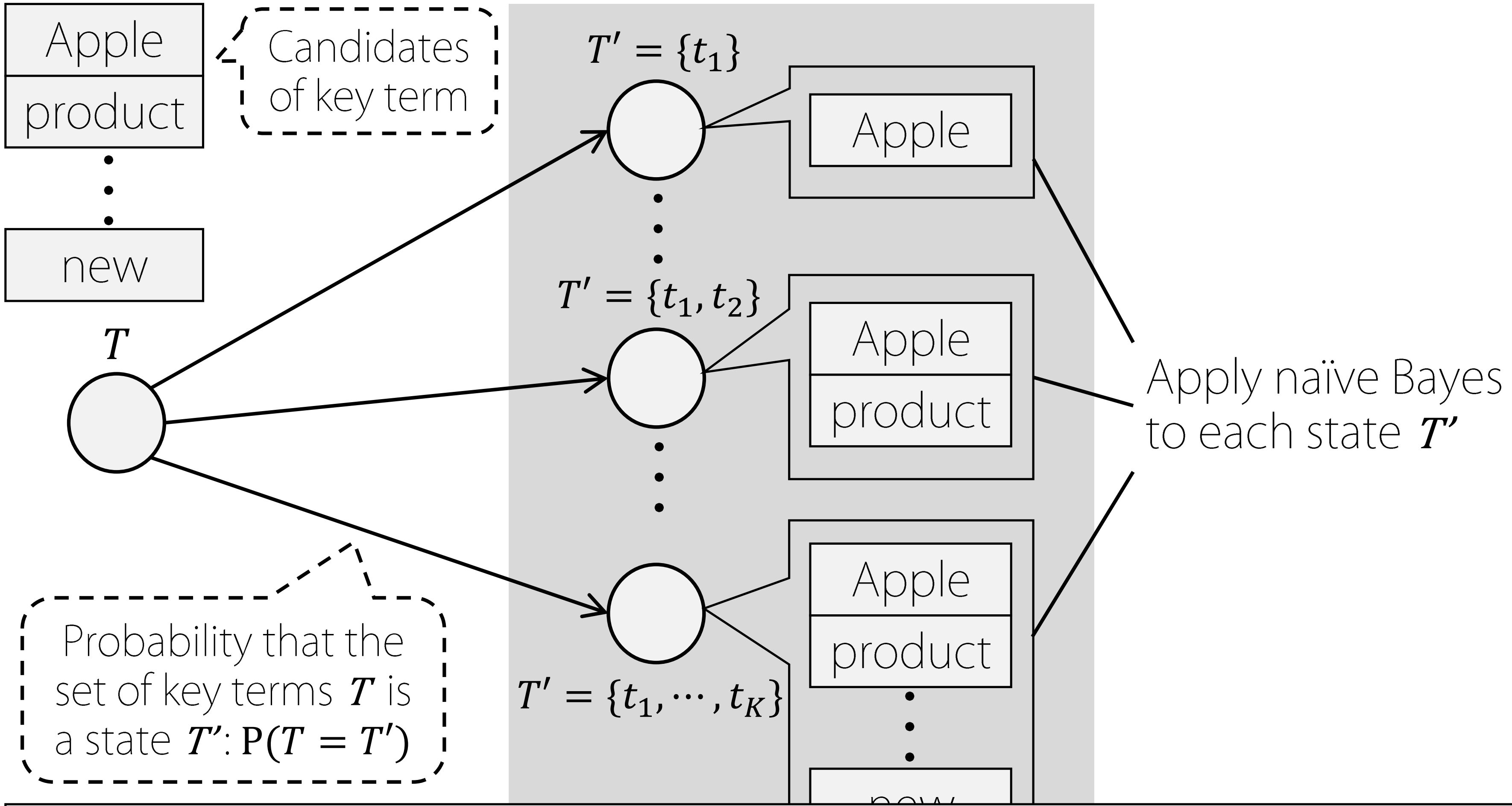
➡ We developed extended naïve Bayes to solve this problem.



Extended naïve Bayes

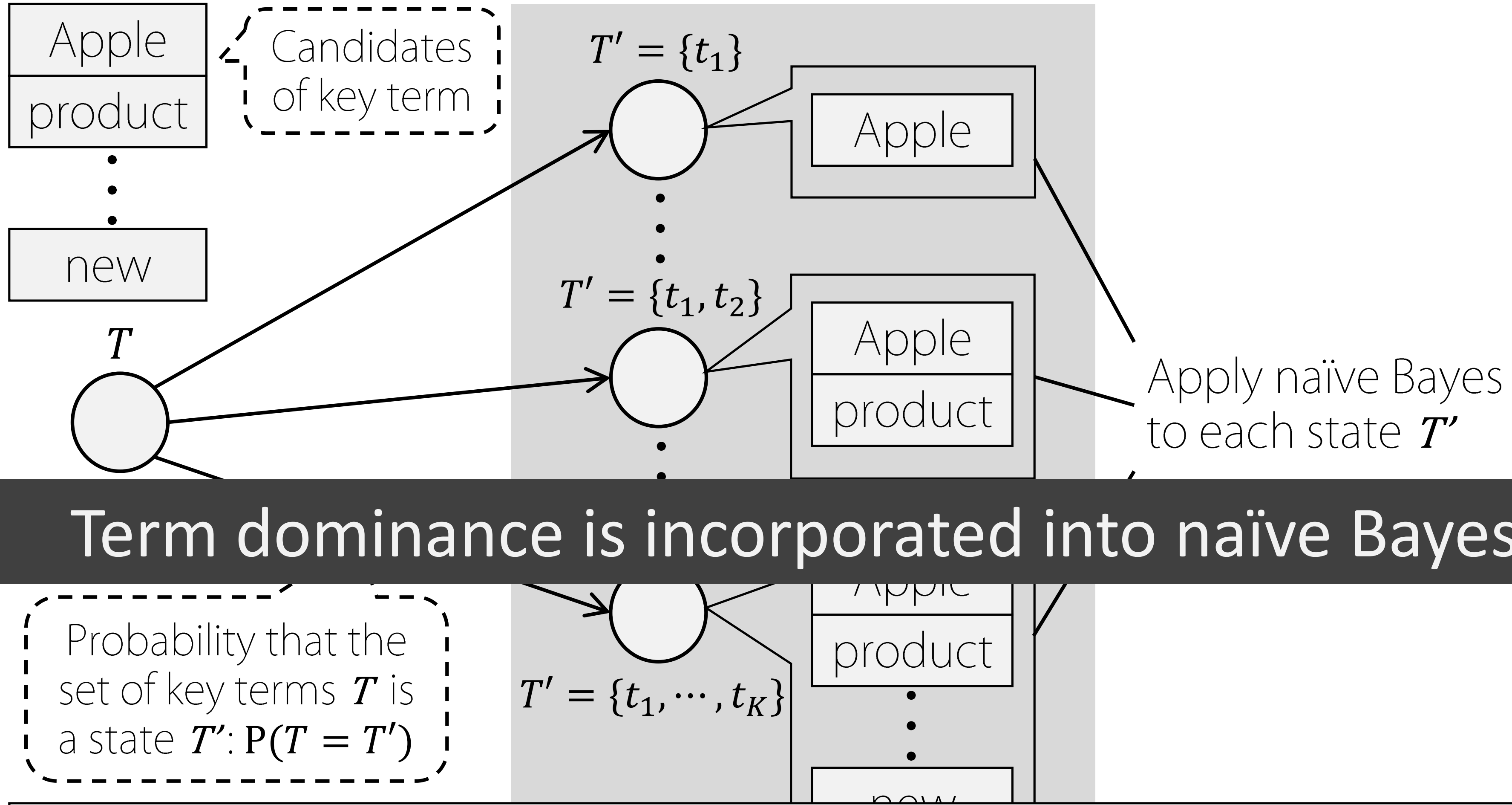


Extended naïve Bayes



$$\sum_{T'} P(c|T')\, P(T = T') = \frac{\prod_k \left(P(t_k \in T) P(c|t_k) + \left(1 - P(t_k \in T)\right) P(c)\right)}{P(c)^{K-1}}$$

Extended naïve Bayes



Term dominance is incorporated into naïve Bayes

$$\sum_{T'} P(c|T')\, P(T = T') = \frac{\prod_k \big(P(t_k \in T)P(c|t_k) + \big(1 - P(t_k \in T)\big)P(c)\big)}{P(c)^{K-1}}$$

Experiments on short text sim datasets

[Datasets] Four datasets derived from word similarity datasets using dictionary

[Comparative methods] Original ESA [Gabrilovich07], ESA with 16 parameter settings

[Metrics] Spearman's rank correlation coefficient

ESA with well-adjusted parameter is superior to our method for "clean" texts.

Method	Pilot	MC	RG	WS
ESA				
KEY-A-L (ESA-same)	0.733	0.777	0.681	0.506
KEY-A-L-COS	0.824	0.826	0.727	0.542
KEY-A-logL	0.823	0.754	0.690	0.571
KEY-A-logL COS	0.797	0.814	0.710	0.559
KEY-logA-L	0.771	0.814	0.626	0.447
KEY-logA-L COS	0.820	0.856	0.650	0.528
KEY-logA-logL	0.866	0.840	0.713	0.505
KEY-logA-logL COS	0.785	0.866	0.706	0.553
IDF-A-L	0.737	0.893	0.790	0.392
IDF-A-L-COS	0.886	0.835	0.791	0.523
IDF-A-logL	0.845	0.869	0.778	0.509
IDF-A-logL-COS (ESA-adjusted)	0.885	0.894	0.806	0.569
IDF-logA-L	0.692	0.746	0.694	0.364
IDF-logA-L-COS	0.856	0.840	0.768	0.505
IDF-logA-logL	0.838	0.838	0.737	0.484
IDF-logA-logL-COS	0.883	0.897	0.784	0.578
Original ESA	0.797	0.833	0.698	0.562
Our method	0.857	0.840	0.717	0.573

Tweet clustering

K-means clustering using the vector of related entities for measuring distance

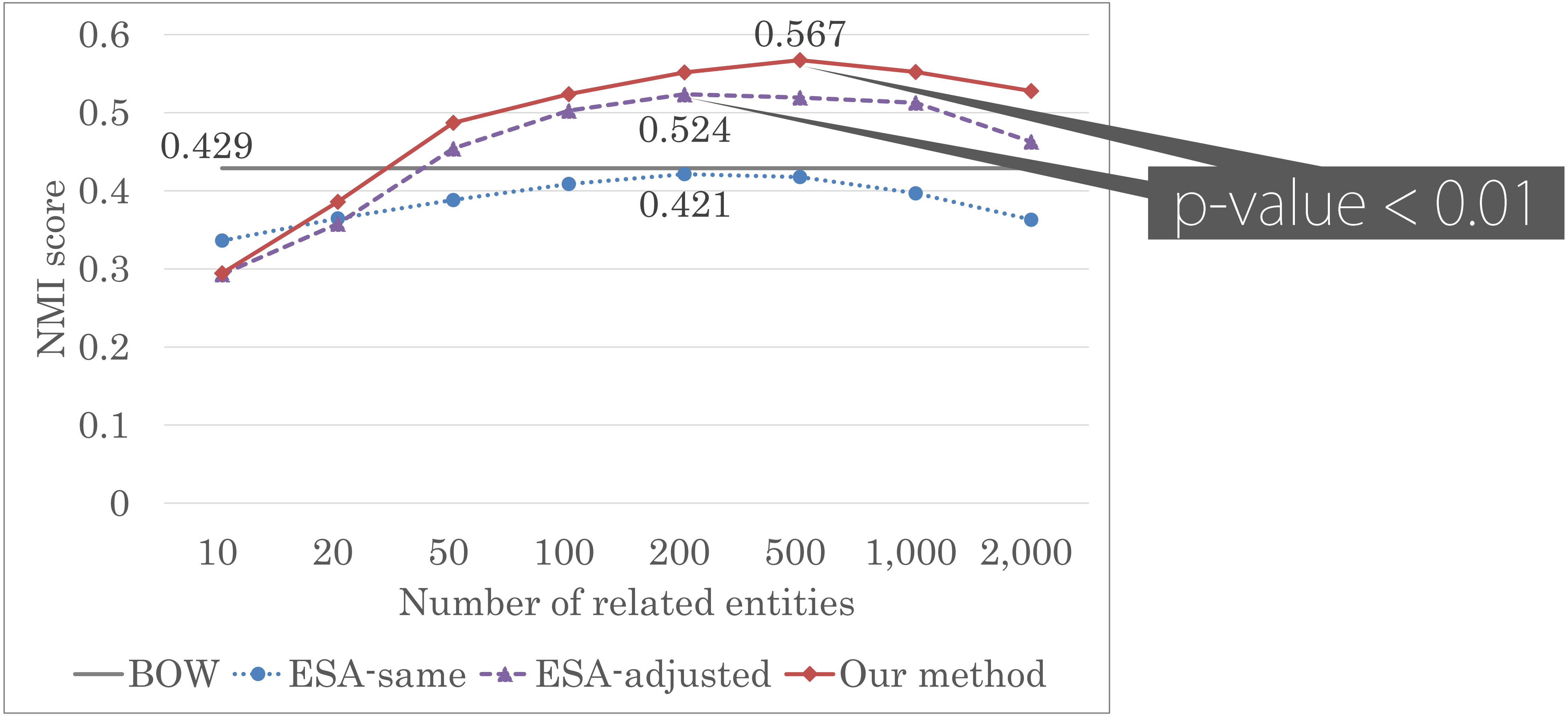
[Dataset] 12,385 tweets including 13 topics

#MacBook (1,251)	#Silverlight (221)	#VMWare (890)
#MySQL (1,241)	#Ubuntu (988)	#Chrome (1,018)
#NFL (1,044)	#NHL (1,045)	#NBA (1,085)
#MLB (752)	#MLS (981)	#UFC (991)
#NASCAR (878)		

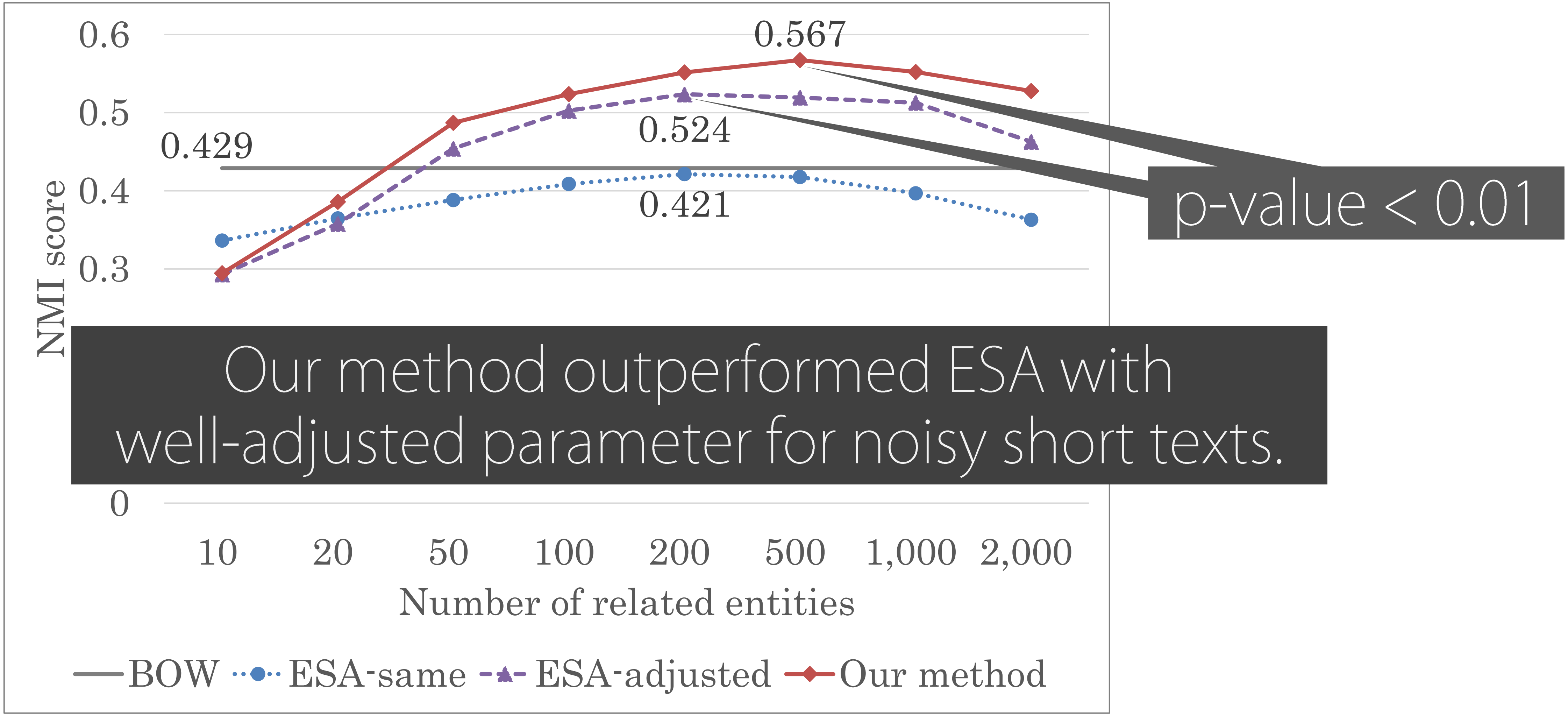
[Comparative methods] Bag-of-words (BOW), ESA with the same parameter, ESA with well-adjusted parameter

[Metric] Average of Normalized Mutual Information (NMI), 20 runs

Results


0.6
0.5
0.4
0.3
0.2
0.1
0
NMI score
0.567
0.524
0.429
0.421
p-value < 0.01
10
20
50
100
200
500
1,000
2,000
Number of related entities
BOW
ESA-same
ESA-adjusted
Our method

Results


0.6
0.5
0.4
0.3
0
0.567
0.524
0.429
0.421
NMI score
10
20
50
100
200
500
1,000
2,000
Number of related entities
BOW
ESA-same
ESA-adjusted
Our method
p-value < 0.01
Our method outperformed ESA with well-adjusted parameter for noisy short texts.

Conclusion

We proposed extended naïve Bayes to derive related Wikipedia entities given a real world noisy short text.

[Future work]

Tackle multilingual short texts

Develop applications of the method