Soft Pattern Matching: Toward Runtime Verification of NLP Systems

Masaki Waga

Kyoto University & National Institute of Informatics

SAIV 2025, 21st July 2025

Based on H. Deguchi, G. Kamoda, Y. Matsushita, C. Taguchi, K. Suenaga, M. Waga, and S. Yokoi "A Soft and Fast Pattern Matcher for Billion-Scale Corpus Searches." ICLR 2025

Personal Scientific Background

- Mainly working on CPS verification
  - Particularly, runtime verification, e.g., monitoring & testing

White-box system model is unnecessary!

- Some Keywords:
  - Signal Temporal Logic
    - Extension of LTL for continuous signals
  - (Timed) Pattern Matching
    - Extract relevant parts of log wrt. formal spec.

w/ quantitative semantics based on distance

LLM Era

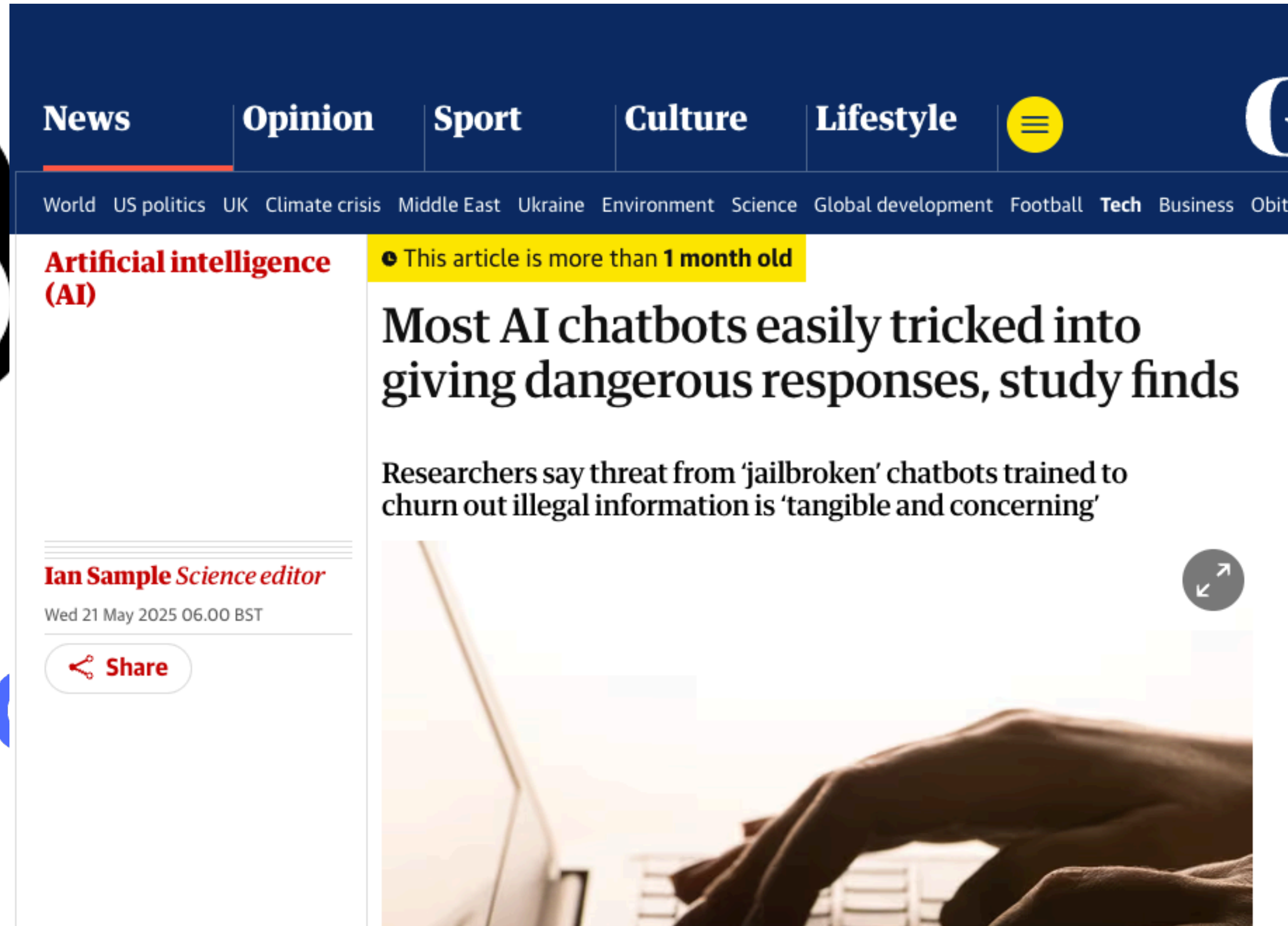
Gemini

Claude

deepseek

https://www.theguardian.com/technology/2025/may/21/most-ai-chatbots-easily-tricked-into-giving-dangerous-responses-study-finds

LLM Era



News | Opinion | Sport | Culture | Lifestyle

World | US politics | UK | Climate crisis | Middle East | Ukraine | Environment | Science | Global development | Football | **Tech** | Business | Obitu

Artificial intelligence (AI)

This article is more than **1 month old**

Most AI chatbots easily tricked into giving dangerous responses, study finds

Researchers say threat from 'jailbroken' chatbots trained to churn out illegal information is 'tangible and concerning'

Ian Sample *Science editor*

Wed 21 May 2025 06.00 BST

Share

https://www.theguardian.com/technology/2025/may/21/most-ai-chatbots-easily-tricked-into-giving-dangerous-responses-study-finds

LLM Era

News Opinion Sport Culture Lifestyle

World US politics UK Climate crisis Middle East Ukraine Environment Science Global development Football Tech Business Obitu

Artificial intelligence (AI)

This article is more than **1 month old**

Most AI chatbots easily tricked into giving dangerous responses, study finds

Researchers say threat from 'jailbroken' chatbots trained to churn out illegal information is 'tangible and concerning'

Ian Sample *Science editor*

Wed 21 May 2025 06.00 BST

Share

Q. What can we, formal method people, do?

https://www.theguardian.com/technology/2025/may/21/most-ai-chatbots-easily-tricked-into-giving-dangerous-responses-study-finds

Observations on LLM

Similar in CPS verification

- System is extremely huge
  - Fully exhaustive verification is likely impossible
- Inputs and outputs are mainly natural language texts
  - Writing spec. w/ conventional logics seems hard
  - Similarity of words should be included in the semantics
    - e.g. "My son is big" $\models \varphi \iff$ "My son is large" $\models \varphi$

Review: Signal Temporal Logic

Continuous-time version of LTL + distance from threshold

$$\varphi ::= \mu > 0 \mid \varphi \land \varphi \mid \neg\varphi \mid \varphi \mathcal{U}_I \varphi$$

Distance from the threshold

Qualitative (Boolean) Semantics

$$\sigma, t \models \mu > 0 \iff \mu(\sigma(t)) > 0$$
$$\sigma, t \models \varphi_1 \land \varphi_2$$
$$\iff \sigma, t \models \varphi_1 \text{ and } \sigma, t \models \varphi_2$$
$$\sigma, t \models \neg\varphi \iff \sigma, t \not\models \varphi$$
$$\vdots$$

Quantitative (Robust) Semantics

$$\rho(\sigma, t, \mu > 0) = \mu(\sigma(t))$$
$$\rho(\sigma, t, \varphi_1 \land \varphi_2)$$
$$= \min\{\rho(\sigma, t, \varphi_1), \rho(\sigma, t, \varphi_2)\}$$
$$\rho(\sigma, t, \neg\varphi) = -\rho(\sigma, t, \varphi)$$
$$\vdots$$

Review: Signal Temporal Logic

Continuous-time version of LTL + distance from threshold

$$\varphi ::= \mu > 0 \mid \varphi \land \varphi \mid \neg\varphi \mid \varphi \mathcal{U}_I \varphi$$

Distance from the threshold

Qualitative (Boolean) Semantics

$$\sigma, t \models \mu > 0 \iff \mu(\sigma(t)) > 0$$
$$\sigma, t \models \varphi_1 \land \varphi_2$$
$$\iff \sigma, t \models \varphi_1 \text{ and } \sigma, t \models \varphi_2$$
$$\sigma, t \models \neg\varphi \iff \sigma, t \not\models \varphi$$
$$\vdots$$

Quantitative (Robust) Semantics

$$\rho(\sigma, t, \mu > 0) = \mu(\sigma(t))$$
$$\rho(\sigma, t, \varphi_1 \land \varphi_2)$$
$$= \min\{\rho(\sigma, t, \varphi_1), \rho(\sigma, t, \varphi_2)\}$$
$$\rho(\sigma, t, \neg\varphi) = -\rho(\sigma, t, \varphi)$$
$$\vdots$$

Review: Signal Temporal Logic

Continuous-time version of LTL + distance from threshold

$$\varphi ::= \mu > 0 \mid \varphi \land \varphi \mid \neg\varphi \mid \varphi \mathcal{U}_I \varphi$$

Distance from the threshold

Qualitative (Boolean) Semantics

$$\sigma, t \models \mu > 0 \iff \mu(\sigma(t)) > 0$$

$$\sigma, t \models \varphi_1 \land \varphi_2$$

$$\iff \sigma, t \models \varphi_1 \text{ and } \sigma, t \models \varphi_2$$

$$\sigma, t \models \neg\varphi \iff \sigma, t \not\models \varphi$$

$$\vdots$$

Quantitative (Robust) Semantics

$$\rho(\sigma, t, \mu > 0) = \mu(\sigma(t))$$

$$\rho(\sigma, t, \varphi_1 \land \varphi_2)$$

$$= \min\{\rho(\sigma, t, \varphi_1), \rho(\sigma, t, \varphi_2)\}$$

$$\rho(\sigma, t, \neg\varphi) = -\rho(\sigma, t, \varphi)$$

$$\vdots$$

Review: Signal Temporal Logic

Continuous-time version of LTL + distance from threshold

$$\varphi ::= \mu > 0 \mid \varphi \land \varphi \mid \neg\varphi \mid \varphi \mathcal{U}_I \varphi$$

Distance from the threshold

Q. Good "distance" for natural languages?

Qualitative (Boolean) Semantics

$$\sigma, t \models \mu > 0 \iff \mu(\sigma(t)) > 0$$

$$\sigma, t \models \varphi_1 \land \varphi_2$$

$$\iff \sigma, t \models \varphi_1 \text{ and } \sigma, t \models \varphi_2$$

$$\sigma, t \models \neg\varphi \iff \sigma, t \not\models \varphi$$

$$\vdots$$

Quantitative (Robust) Semantics

$$\rho(\sigma, t, \mu > 0) = \mu(\sigma(t))$$

$$\rho(\sigma, t, \varphi_1 \land \varphi_2)$$

$$= \min\{\rho(\sigma, t, \varphi_1), \rho(\sigma, t, \varphi_2)\}$$

$$\rho(\sigma, t, \neg\varphi) = -\rho(\sigma, t, \varphi)$$

$$\vdots$$

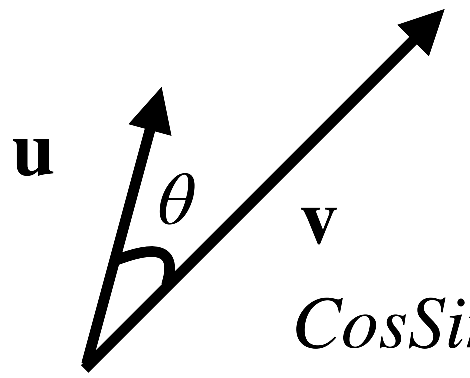
Our Choice: Word Embedding + Cosine Similarity

Word Embedding: Encode words as vectors

- $E \colon \mathrm{Word} \to \mathbb{R}^N$
- Similar usage ≈ similar directions
  - e.g. "He is from Croatia" vs. "He is from Japan"

Cosine Similarity: Measure the similarity by directions

$$CosSim(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$$



$CosSim(\mathbf{u}, \mathbf{v}) = \cos\theta$

Our Choice: Word Embedding + Cosine Similarity

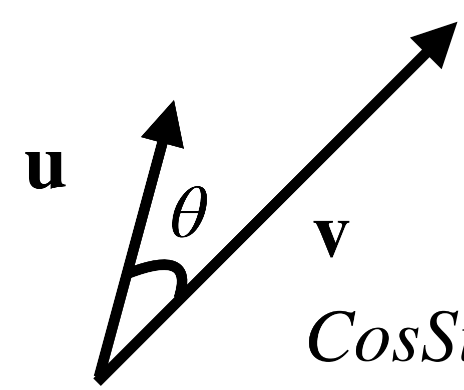
Word Embedding: Encode words as vectors

- $E : \text{Word} \to \mathbb{R}^N$
- Similar usage ≈ similar directions
  - e.g. “He is from Croatia” vs. “He is from Japan”

He is from ?

Cosine Similarity: Measure the similarity by directions

$$CosSim(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$$



$CosSim(\mathbf{u}, \mathbf{v}) = \cos\theta$

Soft Pattern Matching

[Contribution]

String matching + pointwise comparison w/ $CosSim$ and E

Inputs

Typically huge

- Target words: $w = w_1 w_2 \ldots w_n \in \mathrm{Word}^*$
- Pattern words: $p = p_1 p_2 \ldots p_m \in \mathrm{Word}^*$
- Threshold: $\alpha \in \mathbb{R}$

Typically short

Output

Similarity is high
for each word in pattern

- $Match_\alpha(w, p)$ s.t. $i \in Match_\alpha(w, p)$ iff.
  $\forall j \in \{1, \ldots, m\}\ CosSim(E(w_{i+j-1}), E(p_j)) > \alpha$

Example of Soft Pattern Matching

Inputs

- Target w

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}
we	talk	like	we	know	what	we're	talking	about	when	we	talk	about	love

- Pattern: p

p_1	p_2	p_3
we	talk	about

Output

- $Match_{\alpha}(w, p) = \{7, 11\}$

w_7	w_8	w_9
we're	talking	about
≈	≈	=
p_1	p_2	p_3
we	talk	about

w_{11}	w_{12}	w_{13}
we	talk	about
=	=	=
p_1	p_2	p_3
we	talk	about

Indexing-based Algorithm for Soft Pattern Matching

[Contribution]

Offline indexing from target & Online operation on indices

Assumption: Many patterns for single (huge) target

- Reasonable for, e.g. analysis of training data

Note: Algorithm for a stream of target is also available

- Based on Quick Search for string matching

Step 0: Indexing from target

Target w

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}
we	talk	like	we	know	what	we're	talking	about	when	we	talk	about	love

Map each word in w
to its positions

we	{1, 4, 11}
talk	{2, 12}
like	{3}
know	{5}
what	{6}
we're	{7}
talking	{8}
about	{9, 13}
when	{10}
love	{14}

Step 0: Indexing from target

Target w

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}
we	talk	like	we	know	what	we're	talking	about	when	we	talk	about	love

Map each word in w to its positions

we	{1, 4, 11}
talk	{2, 12}
like	{3}
know	{5}
what	{6}
we're	{7}
talking	{8}
about	{9, 13}
when	{10}
love	{14}

Step 1: Compute Pointwise Matching Positions

Idea: "Expand" words in pattern and take union

Pattern: p

p_1	p_2	p_3
we	talk	about

$\{q \mid CosSim(E(q), E(p_1)) > \alpha\}$
= {we, we're, …}

$Pos(p_1, \alpha) = Pos(\text{we}) \cup Pos(\text{we're}) \cup \cdots$
= {1, 4, 7, 11}

we	{1, 4, 11}
talk	{2, 12}
like	{3}
know	{5}
what	{6}
we're	{7}
talking	{8}
about	{9, 13}
when	{10}
love	{14}

Step 1: Compute Pointwise Matching Positions

Idea: “Expand” words in pattern and take union

Pattern: p

p_1	p_2	p_3
we	talk	about

$\{q \mid CosSim(E(q), E(p_1)) > \alpha\}$
= {we, we’re, …}

$Pos(p_1, \alpha) = Pos(\text{we}) \cup Pos(\text{we're}) \cup \cdots$
= {1, 4, 7, 11}

we	{1, 4, 11}
talk	{2, 12}
like	{3}
know	{5}
what	{6}
we’re	{7}
talking	{8}
about	{9, 13}
when	{10}
love	{14}

Step 2: Merge Pointwise Matching Positions by Shift + Intersection

$$Match_\alpha(w,p) = \bigcap_j \{i - j + 1 \mid i \in Pos(p_j, \alpha)\}$$

Pattern: p

p_1	p_2	p_3
we	talk	about

$Pos(p_1, \alpha)$ = {1, 4, 7, 11}
$Pos(p_2, \alpha)$ = {2, 8, 12}
$Pos(p_3, \alpha)$ = {9, 13}
→ $Match_\alpha(w,p) = \{7,11\}$

we	{1, 4, 11}
talk	{2, 12}
like	{3}
know	{5}
what	{6}
we're	{7}
talking	{8}
about	{9, 13}
when	{10}
love	{14}

Step 2: Merge Pointwise Matching Positions by Shift + Intersection

Match each pattern word

$$Match_{\alpha}(w,p) = \bigcap_{j} \{i - j + 1 \mid i \in Pos(p_j, \alpha)\}$$

Pattern: p

p_1	p_2	p_3
we	talk	about

$Pos(p_1, \alpha)$ = {1, 4, 7, 11}
$Pos(p_2, \alpha)$ = {2, 8, 12}
$Pos(p_3, \alpha)$ = {9, 13}
$\rightarrow Match_{\alpha}(w,p) = \{7,11\}$

we	{1, 4, 11}
talk	{2, 12}
like	{3}
know	{5}
what	{6}
we're	{7}
talking	{8}
about	{9, 13}
when	{10}
love	{14}

Step 2: Merge Pointwise Matching Positions by Shift + Intersection

Match each pattern word

Compensate the offset

$$Match_\alpha(w,p) = \bigcap_j \{i - j + 1 \mid i \in Pos(p_j, \alpha)\}$$

Pattern: p

p_1	p_2	p_3
we	talk	about

$Pos(p_1, \alpha)$ = {1, 4, 7, 11}

$Pos(p_2, \alpha)$ = {2, 8, 12}

$Pos(p_3, \alpha)$ = {9, 13}

→ $Match_\alpha(w,p) = \{7,11\}$

we	{1, 4, 11}
talk	{2, 12}
like	{3}
know	{5}
what	{6}
we’re	{7}
talking	{8}
about	{9, 13}
when	{10}
love	{14}

SoftMatcha

- Tool for soft pattern matching
  - Works on laptop w/o GPU
- $ pip install softmatcha
- Online demo is available
  - Target: text from Wikipedia

Demo of SoftMatcha

SoftMatcha

Search | Search

wikitext103 (0.1B) | glove-wiki-gigaword-300

Threshold: 0.5

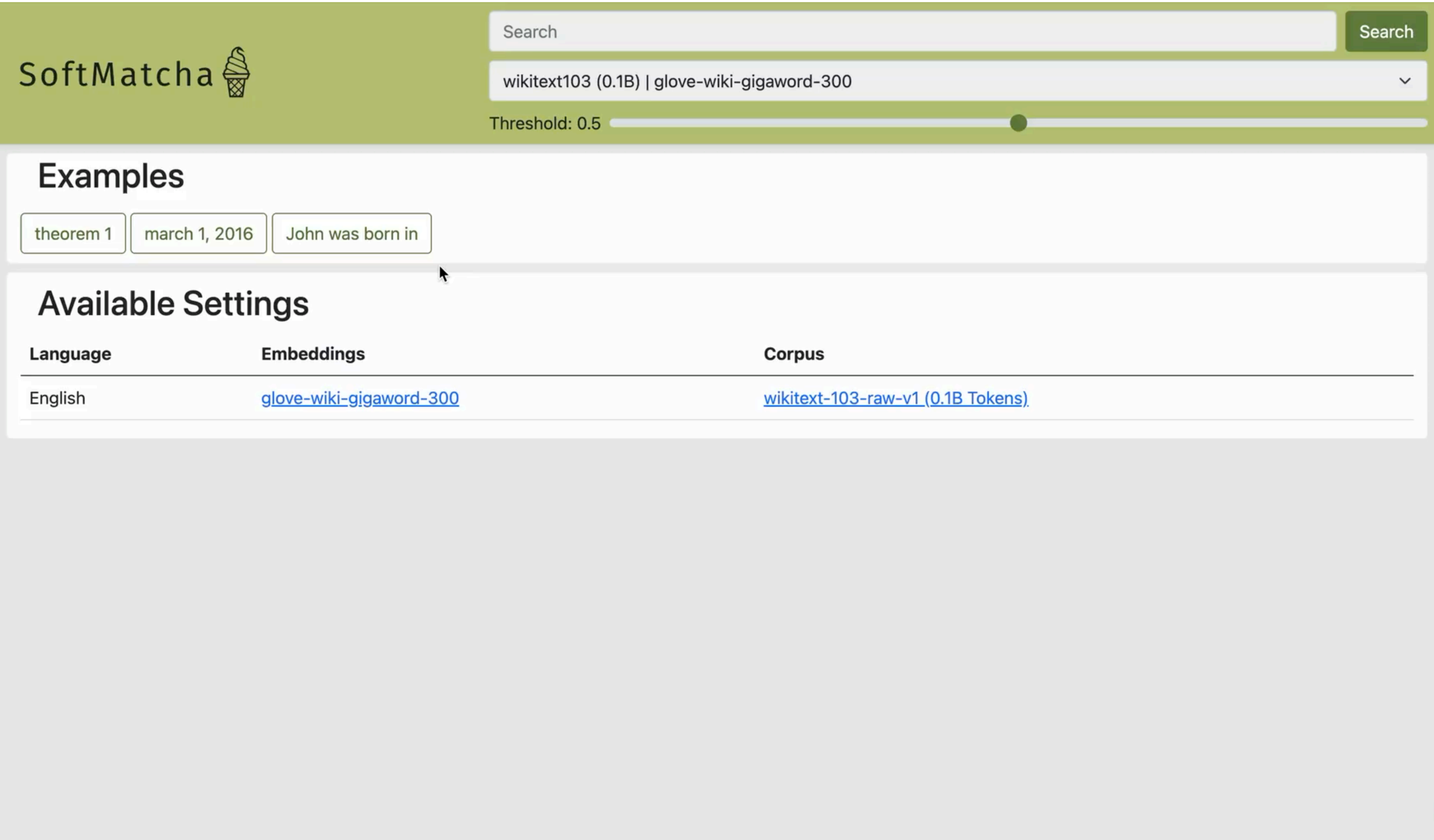
Examples

theorem 1 | march 1, 2016 | John was born in

Available Settings

Language	Embeddings	Corpus
English	glove-wiki-gigaword-300	wikitext-103-raw-v1 (0.1B Tokens)

Demo of SoftMatcha



Demo of SoftMatcha

SoftMatcha

Theorem 1 | Search

wikitext103 (0.1B) | glove-wiki-gigaword-300

Threshold: 0.5

Examples

theorem 1 | march 1, 2016 | John was born in

Results

Hits: 7
Search Time: 0.038 sec

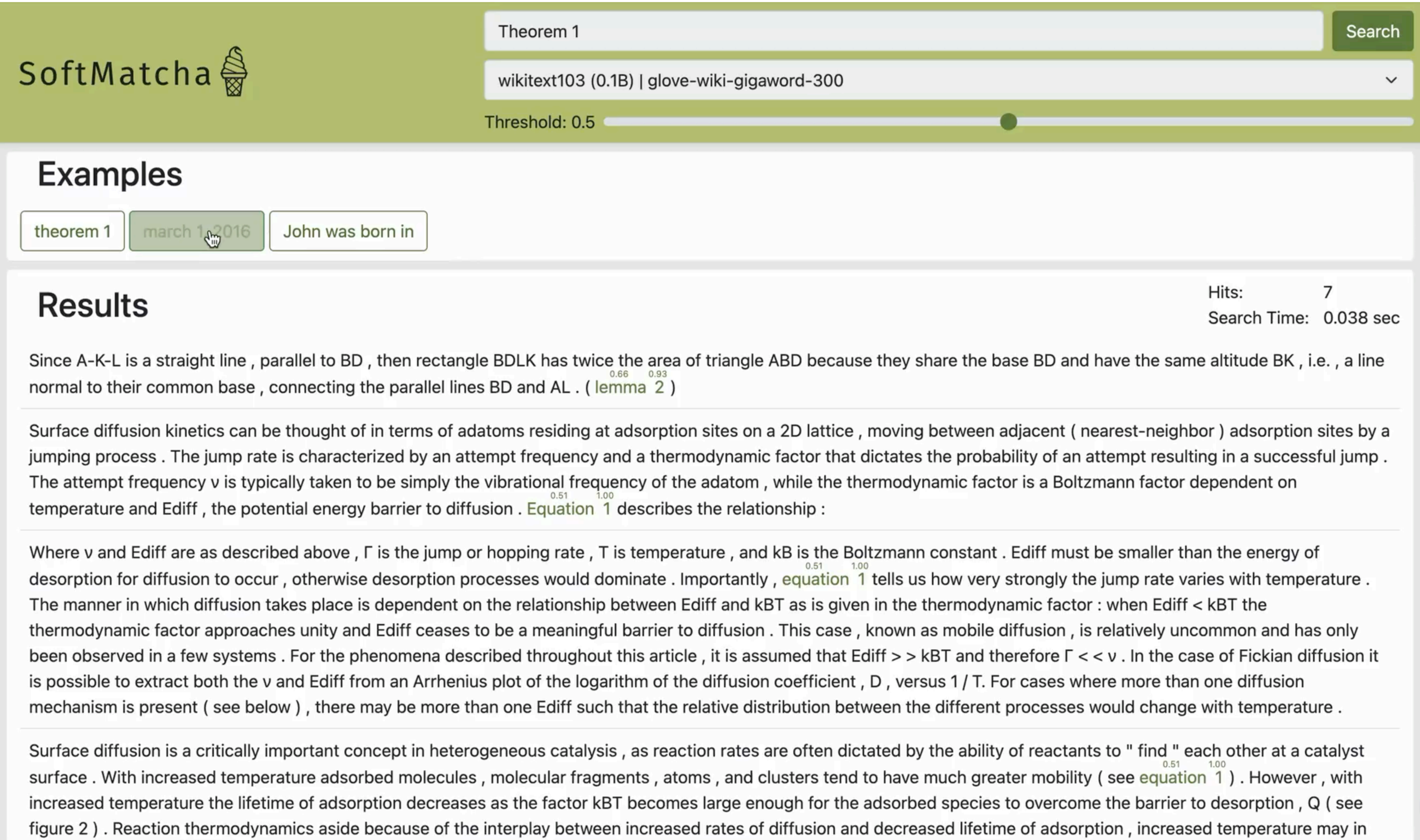
Since A-K-L is a straight line , parallel to BD , then rectangle BDLK has twice the area of triangle ABD because they share the base BD and have the same altitude BK , i.e. , a line normal to their common base , connecting the parallel lines BD and AL . (lemma$^{0.66}$ 2$^{0.93}$)

Surface diffusion kinetics can be thought of in terms of adatoms residing at adsorption sites on a 2D lattice , moving between adjacent (nearest-neighbor) adsorption sites by a jumping process . The jump rate is characterized by an attempt frequency and a thermodynamic factor that dictates the probability of an attempt resulting in a successful jump . The attempt frequency ν is typically taken to be simply the vibrational frequency of the adatom , while the thermodynamic factor is a Boltzmann factor dependent on temperature and Ediff , the potential energy barrier to diffusion . Equation$^{0.51}$ 1$^{1.00}$ describes the relationship :

Where ν and Ediff are as described above , Γ is the jump or hopping rate , T is temperature , and kB is the Boltzmann constant . Ediff must be smaller than the energy of desorption for diffusion to occur , otherwise desorption processes would dominate . Importantly , equation$^{0.51}$ 1$^{1.00}$ tells us how very strongly the jump rate varies with temperature . The manner in which diffusion takes place is dependent on the relationship between Ediff and kBT as is given in the thermodynamic factor : when Ediff < kBT the thermodynamic factor approaches unity and Ediff ceases to be a meaningful barrier to diffusion . This case , known as mobile diffusion , is relatively uncommon and has only been observed in a few systems . For the phenomena described throughout this article , it is assumed that Ediff > > kBT and therefore Γ < < ν . In the case of Fickian diffusion it is possible to extract both the ν and Ediff from an Arrhenius plot of the logarithm of the diffusion coefficient , D , versus 1 / T. For cases where more than one diffusion mechanism is present (see below) , there may be more than one Ediff such that the relative distribution between the different processes would change with temperature .

Surface diffusion is a critically important concept in heterogeneous catalysis , as reaction rates are often dictated by the ability of reactants to " find " each other at a catalyst surface . With increased temperature adsorbed molecules , molecular fragments , atoms , and clusters tend to have much greater mobility (see equation$^{0.51}$ 1$^{1.00}$) . However , with increased temperature the lifetime of adsorption decreases as the factor kBT becomes large enough for the adsorbed species to overcome the barrier to desorption , Q (see figure 2) . Reaction thermodynamics aside because of the interplay between increased rates of diffusion and decreased lifetime of adsorption , increased temperature may in

Demo of SoftMatcha



M. Waga (Kyoto U.)

Demo of SoftMatcha

SoftMatcha

march 1 , 2016 | Search

wikitext103 (0.1B) | glove-wiki-gigaword-300

Threshold: 0.5

Examples

theorem 1 | march 1, 2016 | John was born in

Results

Hits: 7743
Search Time: 0.287 sec

In 2015 , Fey created and produced the television comedy Unbreakable Kimmy Schmidt with fellow 30 Rock-alumnus Robert Carlock . The series stars Ellie Kemper as the titular character who escapes from a doomsday cult and moves to New York . It also stars Fey 's former co-star Jane Krakowski , as well as Tituss Burgess (who had previously appeared in four 30 Rock episodes) and Carol Kane . Although it was originally produced for NBC , it was eventually sold to Netflix and immediately renewed for a second season . The show premiered on March [1.00] 6 [0.81] , [1.00] 2015 [0.67] to critical acclaim .

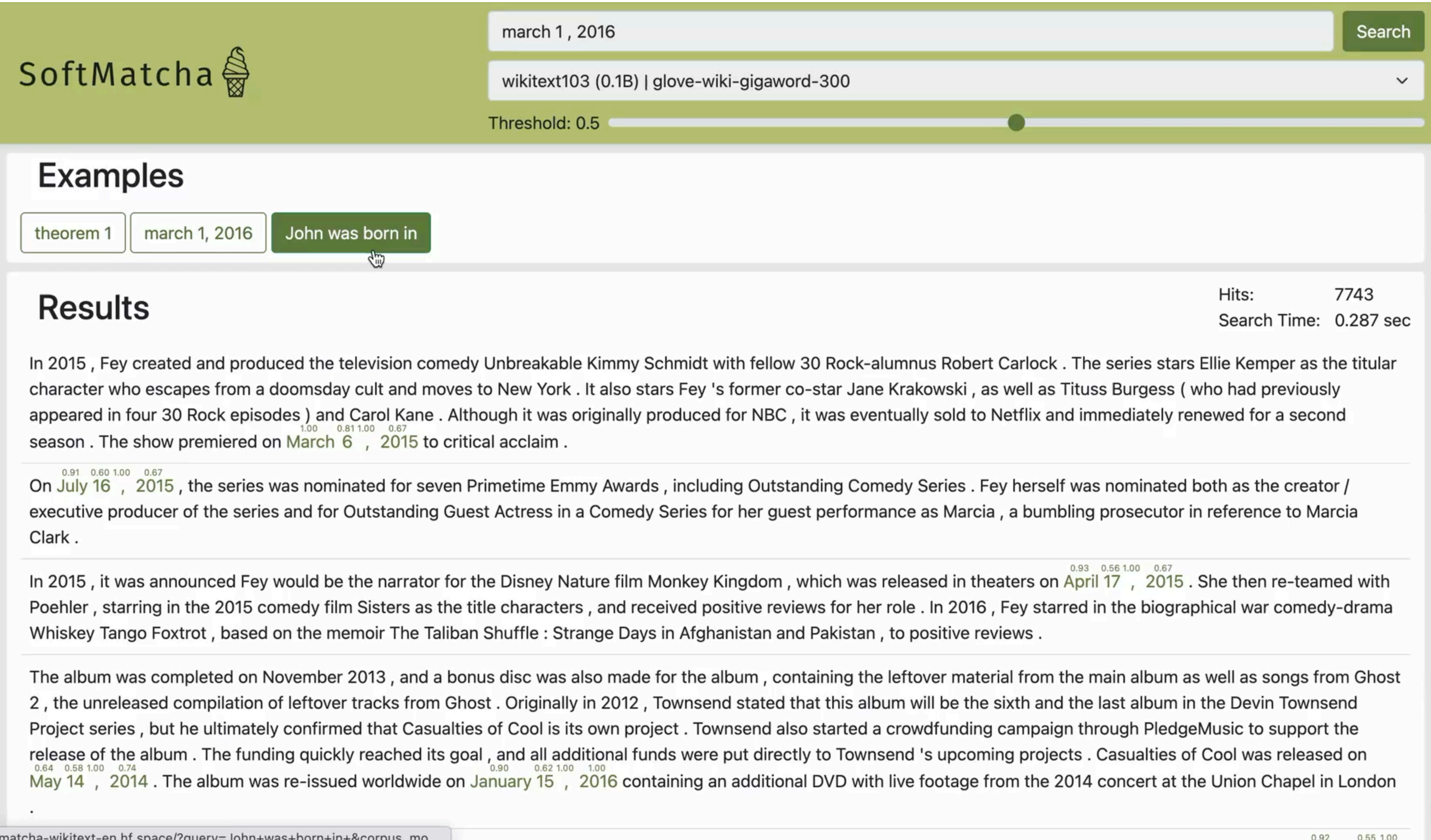
On July [0.91] 16 [0.60] , [1.00] 2015 [0.67] , the series was nominated for seven Primetime Emmy Awards , including Outstanding Comedy Series . Fey herself was nominated both as the creator / executive producer of the series and for Outstanding Guest Actress in a Comedy Series for her guest performance as Marcia , a bumbling prosecutor in reference to Marcia Clark .

In 2015 , it was announced Fey would be the narrator for the Disney Nature film Monkey Kingdom , which was released in theaters on April [0.93] 17 [0.56] , [1.00] 2015 [0.67] . She then re-teamed with Poehler , starring in the 2015 comedy film Sisters as the title characters , and received positive reviews for her role . In 2016 , Fey starred in the biographical war comedy-drama Whiskey Tango Foxtrot , based on the memoir The Taliban Shuffle : Strange Days in Afghanistan and Pakistan , to positive reviews .

The album was completed on November 2013 , and a bonus disc was also made for the album , containing the leftover material from the main album as well as songs from Ghost 2 , the unreleased compilation of leftover tracks from Ghost . Originally in 2012 , Townsend stated that this album will be the sixth and the last album in the Devin Townsend Project series , but he ultimately confirmed that Casualties of Cool is its own project . Townsend also started a crowdfunding campaign through PledgeMusic to support the release of the album . The funding quickly reached its goal , and all additional funds were put directly to Townsend 's upcoming projects . Casualties of Cool was released on May [0.64] 14 [0.58] , [1.00] 2014 [0.74] . The album was re-issued worldwide on January [0.90] 15 [0.62] , [1.00] 2016 [1.00] containing an additional DVD with live footage from the 2014 concert at the Union Chapel in London .

matcha-wikitext-en.hf.space/?query=John+was+born+in+&corpus_mo...

Demo of SoftMatcha



Case Study: finding harmful phrases

Target: corpus in Wikipedia articles (3.4B words)

Pattern: homemade bombs

	Exact Matching	SoftMatcha	Dense vector search
Match Examples	• homemade bombs	• homemade bombs • home-made grenades • homemade missiles	• Article: Survival under atomic attack • Article: Mark 24 nuclear bombs • Article: List of common misconceptions

"Soft" matching but following the pattern

Case Study: finding harmful phrases

Target: corpus in Wikipedia articles (3.4B words)

Pattern: homemade bombs

	Exact Matching	SoftMatcha	Dense vector search
Indexing	685.8 sec	685.8 sec	1036.5 sec
Search	0.005 sec	0.098 sec	0.389 sec

Slower than exact matching but fast

Q. Can we use this for monitoring/testing?

Q. Can we use this for monitoring/testing?

A. Technically Yes.
Practically No so far.

Open Problem: Making Spec.

- Making spec. for LLM seems tricky
  - The application is too wide
  - The expected behavior highly depends on the context

Somewhat similar to perception component in automated driving

- Prohibiting inappropriate words is possible, but can be done more naively

- Perhaps inferring component-level spec from system-level spec?

Conclusions

- We can make pattern matching “soft” w/ a distance
  - Word embedding + cosine similarity
- Indexing-based approach can handle billion-scale corpus
- In the context of RV, making spec. is one major open problem

Conclusions

- We can make pattern matching “soft” w/ a distance
  - Word embedding + cosine similarity
- Indexing-based approach can handle billion-scale corpus
- In the context of RV, making spec. is one major open problem

Any ideas are welcome!