Blooming Onion: efficient deduplication through approximate membership testing

Ondřej Herman

Deduplication

- Large collections of text contain duplicates.
- Humans like to copy.
  - Boilerplate
  - Spam
  - Copy & Paste
  - Headers, Footers
  - Processing artifacts
- Undesirable in downstream applications.

Onion (One Instance Only)

- Deduplication utility
- Based on overlapping n-grams, shingles.
- Discards paragraphs containing a large proportion of previously seen n-grams.
  - 7-grams, over 50 %

	The	**quick**	**brown**	**fox**	**.**		**The**	**quick**	**red**	**fox**	**.**	
	The	**quick**					**The**	**quick**				
		quick	**brown**					**quick**	**red**			
			brown	**fox**					**red**	**fox**		
				fox	**.**					**fox**	**.**	

Onion

- Hashes of shingles are stored in an associative array
  - Judy array
  - Hash table in newer versions

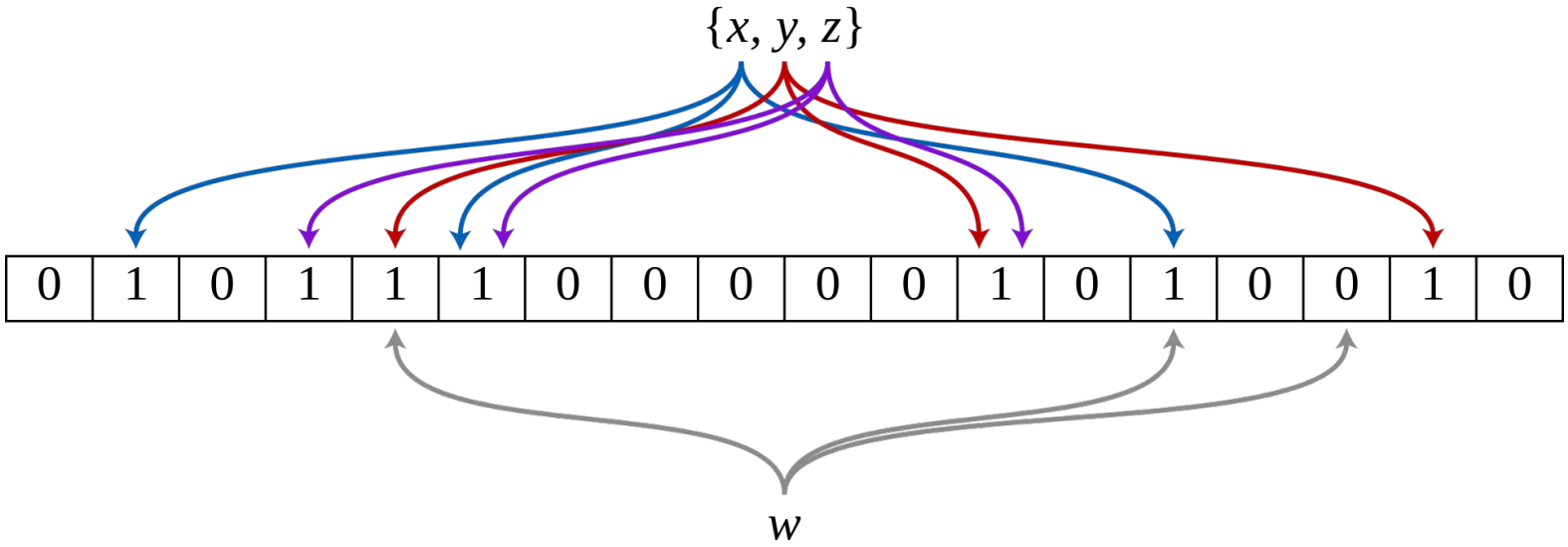
Blooming Onion

Gluten Free Vegan Baked Blooming Onion

https://petiteallergytreats.com/gluten-free-vegan-baked-blooming-onion/

Blooming Onion

- Same principle as Onion, but use a Scalable Bloom Filter to store the shingles.
- Written in Rust, under 150 SLOC.



Blooming Onion

- No false negatives.
  - More strict than the exact variant – more text might be identified as duplicate.
- The false positive rate for the Scalable Bloom Filter is set to 1 %.

Evaluation

Susanne corpus, repeated 20 times (100 MB, 190 k lines, 97.5 % duplicate)

	Runtime	Max RSS
Blooming Onion	1.94 s	3608 MB
Onion	1.71 s	30616 MB

7 days of the JSI Newsfeed Corpus (13 GB, 876 k lines, 64 % duplicate)

	Runtime	Max RSS
Blooming Onion	720.6 s	271.6 MB
Onion	491.3 s	2367.2 MB

Conclusion

- I wrote a prototype deduplication tool, which is 25 % slower than Onion, but requires only 10 % of the memory.