LEXALYTICS

WHITE PAPER

OCR errors skew the text analytics used for making good decisions.

Salience 7 gives you the ability to fix OCR errors after they are made but before doing any text analytics.

Post-OCR Error Correction in Salience 7

In a perfect world, you would stop reading this right now. Sadly, the optical character recognition (OCR) engines that create those electronic documents you need for your business intelligence make errors. And these errors skew the text analytics you use for making good decisions. So, do read on.

Odds are you have no control over the OCR processing. There is no one you can pester or plead with about scanning the documents more carefully. By the time you receive a document, the damage is done. The good news is that *Salience 7* gives you the ability to fix OCR errors after they are made but before doing any text analytics.

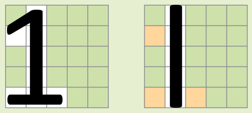
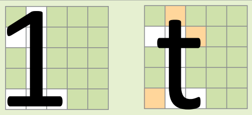
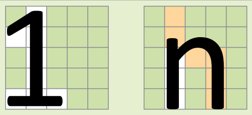
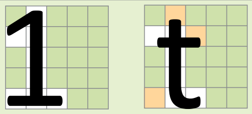
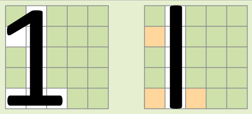
But isn't OCR error correction "just" spell-checking? In principle, yes. In practice, no. Spellcheckers are looking for typing errors. **Lexalytics' OCR error correction is tuned to catch mistakes that happen because two letters are visually similar**. For example, an OCR engine might confuse "m" for "iii" because both glyphs contain three vertical strokes. An ordinary spellchecker would be unlikely to guess that three letters ("iii") should be replaced by just one ("m"). *Salience 7* catches this error.

UNDERSTANDING HOW OCR ERROR CORRECTION WORKS

You may not realize this, but you already understand how OCR error correction works. Consider the phrase "**a whi1e**". This is just the sort of error an OCR engine might make. Which of the following revisions do you think is best? 1) **a**whi**l**e. 2) a whi**t**e**r**. 3) a whi**n**e. 4) a whi**t**e. 5) a whi**l**e.

Odds are you chose correction 5: "a while." Why? Let's step through your thinking. You probably preferred correction 5 over 1 because correction 1 unnecessarily changed the number of words from two ("a while") to one ("awhile"), but correction 5 kept the number of words the same. You probably also preferred correction 5 over 2 because correction 2 ("a whiter") changed two letters — unnecessarily adding an "r" — while correction 5 changed only one letter. Finally, you probably preferred correction 5 over 3 because "1" and "t" (in correction 5) differ by only 3 pixels (colored peach in *Chart 1*), but "1" and "n" (in correction 3) differ by 6 pixels. The choice finally came down to corrections 4 and 5, which both change the same number of words (0), letters (1), and pixels (3). You probably broke this tie by picking the word "while" over "white" because "while" is more common. In short, the rule you used was "Pick the most common word requiring the fewest changes."

Chart 1 | ***Comparison of Possible OCR Corrections*** *as analyzed by Word Edit Distance, Letter Edit Distance, Pixel Edit Distance and Word Rank*

	Possible Corrections	|W| Word Edit Distance	|L| Letter Edit Distance	|P| Pixel Edit Distance		|R| Word Rank		
1	**a**whi**l**e.	**1**	1	3		–	✗	Too many words changed: 1 vs. 0
2	a whi**t**e**r**.	0	**2**	3		–	✗	Too many letters changed: 2 vs. 1
3	a whi**n**e.	0	1	**6**		–	✗	Too many pixels changed: 6 vs. 3
4	a whi**t**e.	0	1	3		**7**	✗	Less popular: 7th vs. 6th in ranking
5	a whi**l**e.	**0**	**1**	**3**		**6**	✓	**Most popular, fewer changes**

OCR CORRECTION BREAKDOWN OF CHANGES

The kinds of changes you cared about were:

|W|, the word edit distance:
the number of words changed by the correction

|L|, the letter edit distance:
the number of letters changed by the correction

|P|, the pixel edit distance:
the number of pixels changed by the correction

You also had a subjective judgment of how common a word was. This is expressed numerically as:

R, the word rank:
on a scale of 1 (most common) to 20 (least common)

These factors can all be mathematically combined a principled way to guarantee the most likely correction:

$$10^{-R} \times p_W^{|W|} \times p_L^{|L|} \times p_P^{|P|}$$

where p_W, p_L and p_P are the probabilities that the OCR engine mistakenly changed a single word, letter, and pixel, respectively. The best correction will be the one for which the value of this expression is the highest.

Although this expression is what you need, it may not be what you want to deal with, what with its probabilities and exponents and all. As it turns out, it can be converted into a mathematically equivalent form[1] that mirrors how you think about OCR correction:

$$R + s_W \times |W| + s_L \times |L| + s_P \times |P|$$

which adds together a word's rank with all the edit distances (multiplied by scale factors). Now the best correction to use is the one whose value of this expression is lowest. Just like you, it "picks the most common word requiring the fewest changes."

[1] *By taking its negative logarithm, if you really must know...*

GOOD OCR ERROR CORRECTION NEEDS GOOD DICTIONARIES

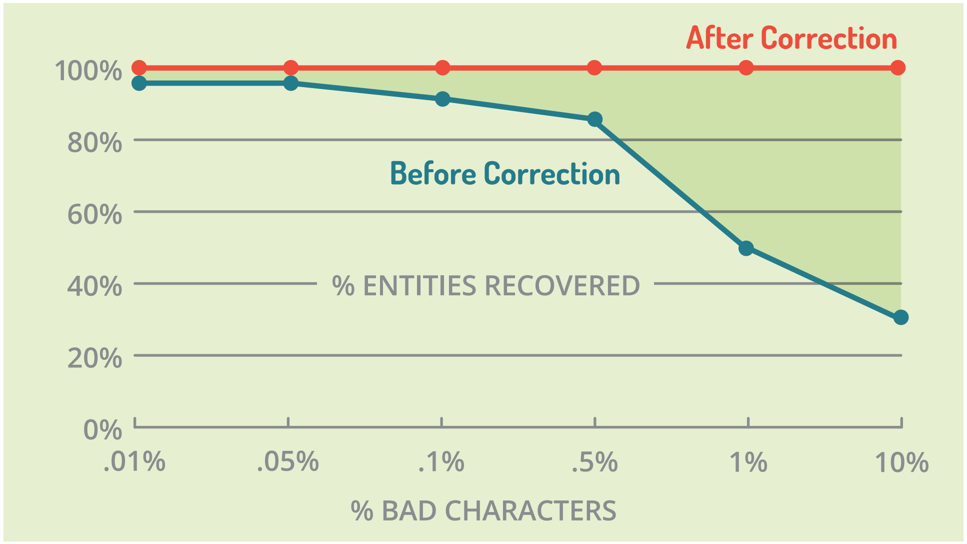
Without good dictionaries, how could it know not to capitalize the "I" in "iPhone"? *Salience 7's* OCR error correction has extensive coverage of common vocabulary and proper names in its built-in dictionaries. It also includes two ways of extending these built-in dictionaries to account for new words. First, ***Salience 7*** **allows you to easily add auxiliary dictionary files to account for your custom vocabulary**. The format of these additional dictionaries is simple: a tab-separated list of custom vocabulary terms and their ranks. For example:

twerk	12
twerked	12
twerking	12

Second, ***Salience 7*** **creates a temporary, on-the-fly dictionary for each document, adding any new words found in the document**. Misspellings of these words within the document are corrected against this dictionary.

We've just described the default behavior of *Salience 7*, which is to correct isolated words. This approach gives good results, but it also means that "a whi1e" will always be corrected to "a while," even for the sentence "I saw a whi1e house." Clearly, "white" would be a better choice in this case. We can get even better error correction by using the other words in the sentence to estimate a contextual rank R_c for a word. In experiments, we found that using contextual rank in place of the ordinary rank R gave a 10% further reduction in OCR errors. A subsequent release of *Salience* will use a deep-learning model to do optional contextual OCR error correction.

Chart 2 | ***Percentage of Entities Recovered*** *Before and After Correction of Bad Characters*



HOW OCR ERROR CORRECTION RECOVERS CORRUPT TEXT ANALYTICS

So, what exactly is the accuracy of *Salience 7's* OCR error correction, anyway? To be blunt, this is not quite the right question to ask. A better question would be how well OCR error correction recovers the text analytics lost to corrupted text.

For example, one of our customers reported that OCR engines hallucinated periods (".") throughout their documents, breaking them into ungrammatical, choppy sentences like this:

> **.Teja.s Securities. Gr.oup, I.nc. is. a full servi.ce b.rokerage. and inves.tment-banking firm based in Au.stin, Texas with bran.ch offices in Atlanta, .Geo.rgia.**

If you're guessing this degraded their text analytics, you're right.

SALIENCE 7 REAL-WORLD AND REAL-WORD TESTS

We tested how *Salience* 7 handled bad sentence breaks by randomly injecting periods into documents, comparing the percentage of entities recovered before and after OCR error correction (see *Chart 2*). Even with an OCR error rate as small as 0.01%, a few entities were still lost to errors before correction. And at the largest error rate of 10%, 65% of the entities were lost! ***Salience 7's*** **non-contextual OCR error correction fixed this, recovering 100% of the entities, whatever the error rate**.

We also tested *Salience 7* on generic OCR errors, randomly corrupting letters in documents to simulate typical OCR errors. Here's a sample of the corrupted text:

> **autilusFootwear is differec,|t. Nautilus pa-ved the way over thelast few years in designing ergonomially designed footwe.ar to p[otecf workers in moreways than J:'ver before.**

Chart 3 | ***Percentage of Entities Recovered*** *Before and After Correction of Bad Characters*

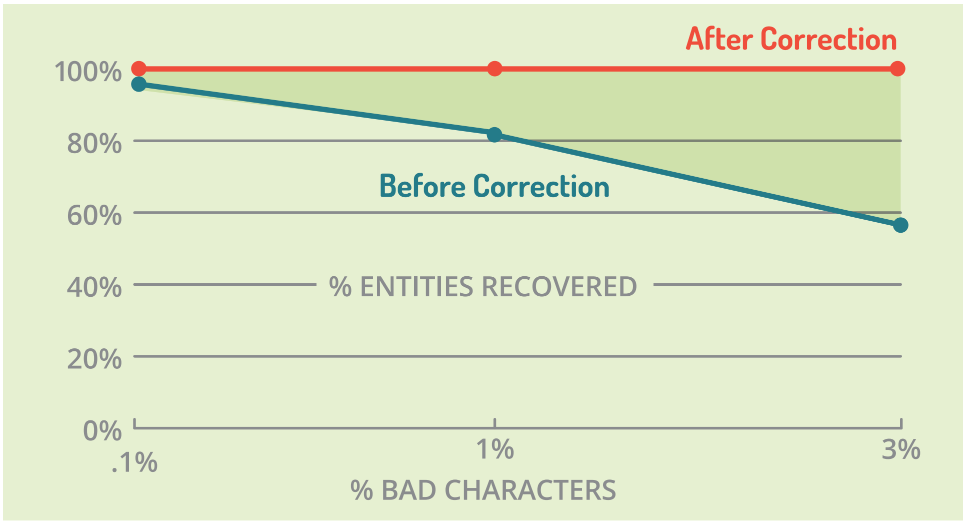


Chart 3 above shows the percentage of entities recovered before and after correction. If you squint a bit, you can see that a few entities were lost even with a tiny OCR error rate of 0.1%. At a modest 1% OCR error rate, a sizable 15% of the entities were lost before correction. And when the error rate was 3% (not atypical for commercial OCR engines), the number of lost entities jumped to 40%. Again, ***Salience 7's* non-contextual OCR error correction fixed this, recovering 100% of the entities at all error rates**.

SUMMARY

OCR technology promises to grant you access to business intelligence that has been locked away inside paper documents. Unfortunately, much of this remains blocked by OCR errors. *Salience 7* unlocks it, allowing you to drive your text analytics from all the documents you have on hand, whether electronic or paper.

Lexalytics processes BILLIONS of text documents every day, GLOBALLY.

We help our clients unlock the full value of their text data through innovative text analytics solutions delivered through the modular Lexalytics Intelligence Platform® and "semi-custom" enterprise applications.

Our solutions combine natural language processing, semi-structured data parsing, and machine learning to reveal context-rich patterns and insights within comments, reviews, contracts, medical files and other complex text documents.

Enterprises and data analysts rely on Lexalytics to improve customer experiences, reduce employee turnover, manage regulatory compliance, improve process automation, and more by gaining deeper insights and greater value from their data.

For more information, visit www.lexalytics.com or call 1-800-377-8036